

# MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA  
TELEPHONE (08) 272 0966

ISCIF  
AGS 10

PRICE:- AUS. \$2.50  
- NZ. \$3.00

\*\*\*\*\* CONTENTS \*\*\*\*\*

CHICKS- is registered for training & soliciting - 100% Y.

All inquiries should be directed to:-  
 THE DIRECTOR, C. I. D., P. O. BOX 10, NEW DELHI.

Printed by :-  
 The ... & ... Printers  
 3rd ...  
 ...

# MICRO-80

**\*\* ABOUT MICRO-80 \*\***

EDITOR: Ian Vagg  
ASSOCIATE EDITORS:  
SOFTWARE: Peter Hartley  
HARDWARE: Edwin Paay

MICRO-80 is the only Australian monthly magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80. It is available by subscription, \$24.00 for 12 months or by mail order at \$2.50 per copy. A cassette containing all the programs in each month's issue is available for an additional \$3.50 or a combined annual subscription to both magazine and cassette is available for \$60.00. Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80 or System 80 and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

**\*\* WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS \*\***

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 or System 80 to earn some extra income is included in every issue.

**\*\* CONTENT \*\***

Each month we publish at least one applications program in Level 1 BASIC, one in Level 2 BASIC and one in DISK BASIC (or disk compatible Level 2). We also publish Utility programs in Level 2 BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

**\*\* COPYRIGHT \*\***

All the material published in this magazine is under copyright. That means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.

**\*\* LIABILITY \*\***

The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

## \*\*\*\*\* EDITORIAL \*\*\*\*\*

## \*\* TANDY VERSUS '80 USERS' GROUPS \*\*

We have had the impression for some time now that the Tandy establishment is unhappy about the existence of TRS-80 Users groups. Nevertheless, individual store managers have often taken it on themselves to foster such groups, even to the extent of staying back late at night to allow their premises to be used as a meeting place. Now, there has been a firm directive from Rydalmere FORBIDDING Tandy stores from allowing Users' groups to meet on their premises. Henceforth, only Tandy Computer Centres are permitted to provide facilities for Tandy Users' Groups. If the manager of the local computer centre does not want to provide that service, that's bad luck, the Users' Group must fend for itself. We know of at least one such Group which has had to find new, non-Tandy premises.

From one point of view, we think this is a pretty sad state of affairs. Tandy Australia is really behaving no differently in this than Radio Shack in the USA. Rather than feeling proud that they have been instrumental in creating a whole new industry, not to mention having brought a great deal of pleasure to thousands of people, it seems that Tandy resents the intrusion of others into their field and would like to keep the whole TRS-80 industry (and sales) to themselves. Acts such as the above only serve to alienate established customers and force them to turn to other suppliers, thus defeating Tandy's purposes anyway.

There is however, a bright side to the picture. There are now well over 1000 System 80 Users. We know that all the TRS-80 Users groups are only too happy to welcome them and certainly System 80 Users would gain a lot from joining a Users' group. Recently, we have spoken to number of System 80 owners and suggested that they might like to attend Users' group meetings. It seems that, where Users' group meetings are held in Tandy stores, many System 80 owners do not like to attend, because they feel they would be intruding. So, once the displaced Users' groups get over the trauma of finding new homes, they might actually thank the Tandy directive because they will almost certainly gain many new members with System 80's.

Whilst we are on the subject of Tandy and its attitudes, elsewhere in this issue, you will find an open letter from Peter Hartley to Tandy Australia concerning its attitude towards repairing modified TRS-80's. Peter, who has a fairly pithy style of writing anyway, was pretty steamed up when he wrote the letter. We'll let you judge whether he is being reasonable or not. It does seem, however, that Tandy Australia might be lagging behind Radio Shack in its policy in this area. When the Radio Shack lower case mod. kit was first released in the USA, for example, Radio Shack charged about \$20 to fit it to a customer's CPU. However, if that CPU had already had a non-Radio Shack lower-case mod kit fitted, then he was charged \$20 to have the existing kit removed and then a further \$20 for the normal fitting fee. That sounds very much like the practice adopted in the case reported by Peter Hartley. We understand that, in the USA, the charge for returning the unit to standard, no longer applies. How about it Tandy, why not follow the same policy here?

Mal Williams of Tandy Australia wrote us a nice letter recently in which he invited us to contact him to check out any "rumours" etc. . Time has not permitted us to do so before this issue went to press but we will certainly offer Mal and Tandy the opportunity to reply to this Editorial and Peter's letter, in the next issue.

## \*\* SYSTEM 80 TO TRS-80 ADAPTOR BOARD \*\*

The adaptor board to provide a printer port and TRS-80 bus from the System 80 has taken MICRO-80 PRODUCTS longer to develop than anticipated, not least because of a faulty Z-80 in the test-bed System 80! Development is now just about complete however and it has successfully driven a printer an Exatron Stringy Floppy and Disk drives via the Tandy expansion interface. Deliveries should start about the end of September. It has been decided to make the expander unit available either in kit form or fully built up form. the kit will come in three stages:- printer only; printer plus ESF/Disk signals ; full kit providing memory refresh to the expansion interface. Each kit will contain the same PC board, the only difference will be in the components supplied (and the price, of course). Full details and prices will be announced in the September issue of MICRO-80.

In the meantime, you can secure early delivery on a \$10.00 deposit, refundable in full if you think the final price is too high (the full-house unit, built-up and tested will certainly be priced at less than \$100 ).

**\*\* MORE ABOUT SYSTEM 80 COMPATABILITY \*\***

Further to our guidelines in last month's issue, for writing System 80 compatible software, now that we have evolved a simple mod. for the System 80 to give software control over the 32 character format, item 1 should be amended to read:-

1. If the largr (32 character) mode is used, then the screen display must look correct in both the 32 CHARACTER MODE and the System 80 NORMAL SIZE BUT DOUBLE-SPACED MODE. This mode should be enabled by using OUT 255,8 and not CHR\$(23).

**\*\* NEW FORMAT \*\***

Once again we have improved the format of MICRO-80. This time to allow us to squeeze more in without incurring higher printing costs. We hope you like the change. Incidentally, at the time of writing, we already have 900 subscribers and the numbers are growing rapidly. Tell your friends about us, the more subscribers we have the better the service we can offer.

**\*\* MELBOURNE HOME COMPUTER SHOW \*\***

The Melbourne Home Computer Show has been cancelled - or in the words of the promoters, postponed indefinitely. This Show was to have taken place from 11-14 September and we had booked a stall. All is not lost, however. The 8'th World Computer Conference is on in Melbourne in the week commencing 13'th October. We have managed to book a stall - No. 169. This will be an enormous exhibition, taking the whole of the Melbourne Exhibition Buildings. We look forward to meeting many of our Victorian readers. More details in the September issue.

- 00000 -

**\*\*\*\*\* '80 USERS' GROUPS \*\*\*\*\***

The following is a list of '80 Users- Groups. If you have a group that is not included here, please let us know about it so that we can publish details. Owners of System 80's are welcome at all the groups.

BRISBANE: CONTACT, Mr. Lance Lawes, 21 Rodney Street, LINDUM, 4178  
Telephone:- home (07) 396 2998; bus. (07) 268 1191 Ext. 15

**MELBOURNE EASTERN-SUBURBS**

CONTACT: Mr. John Fletcher, 89 0677 between 9 - 4  
MEETINGS: 1st Wednesday Tandy Store,(?) 96 Koornang Road,  
of the month Carnegie.  
3rd Wednesday Kingswood College,  
of the month 355 Station St., Box Hill

DARWIN: CONTACT:- Tony Domigan, P.O. BOX 39086, Winnellie, NT. 5789

ADELAIDE: CONTACT:- Rod Stevenson, 51 5241 between 9 - 4.

- 00000 -

**\*\*\*\*\* MARKET PLACE \*\*\*\*\***

Market place is available to any reader who has hardware to dispose of. An entry costs nothing - you pay MICRO-80 \$5 or 5% commission, whichever is the greater - up to a maximum of \$30, after the goods are sold. The commission is calculated on your advertised price.

Near new L1/16K TRS-80, standard monitor includes Personal Finance program. Seldom used. Total catalogue price \$1016 will sell for only \$800.

Mr. Ward, 19 Filmer Ave., PARA HILLS, Adelaide. Tel. (08) 264 6373 after 5.30 p.m.



**\*\* G.T. BASIC \*\*** Part One in a series, by Peter G. Hartley.

#### INTRODUCTION

This series is intended to take you right from being a raw beginner and help you to become an experienced programmer. Because this series is aimed right at YOU, I would appreciate your letters and comments, throughout the series, in order that I can ensure that the articles are what YOU want to see.

Each month I will discuss various commands or functions and, where necessary, comparisons will be drawn between the usage of these in Level I, Level II and Disk Basic. Don't expect to get everything about a particular command or function in a single serving, because this series is aimed at producing what I call G.T. BASIC - which is very highly tuned - and we will revisit sections from time-to-time, with a view to improving our knowledge and understanding as we progress. Later in the series, I will discuss TRSDOS (whatever the current version may be), NEWDOS (primarily because this is the DOS of MY choice, offering the majority of the "extras" that I feel should be "standard") and MICROSOFT LEVEL III BASIC. Where a major difference exists between the TRS-80 and the System '80, this will be documented.

As far as possible I will avoid detailed explanations of the internal workings of the '80 and of the finer points of its 'thinking' processes, although some generalised introduction will be made in the early articles.

The arrangement of the series is neither alphabetical, nor by groupings of related functions but rather in a sequence that I hope will enable the raw beginner to progress in an ordered manner towards the creation of truly useful programs. I trust that the more experienced of our readers will bear with this order of progression?

From time to time you will come across some of MY rules - called G.T. BASIC RULES. These are NOT rules that HAVE to be followed, to make your '80 work. They ARE rules that will lead to smoother running, faster executing programs. Follow them to "fine tune" YOUR BASIC.

#### **\*\* COMMUNICATING WITH THE DUMB '80 \*\***

For the purposes of this series, we will regard the '80 as being a very obedient but incredibly dumb machine. It is possible to communicate with the '80 through the keyboard, using BASIC (Beginners All-purpose Symbolic Instruction Code).

The '80 understands only about 120 words, but it will implicitly obey these (if at all possible) until told to stop. Until you tell it exactly what to do, it is just a very expensive piece of junk.

#### **\*\* DUMB, STUPID \*\***

Because the '80 is really very very stupid, it is unable to make any compensation for human error unless it is taught how to do this. (Of course, the '80 cannot do anything much at all until it is taught how to do it - and that's what programming is all about!) Therefore, it is necessary that our instructions are always totally clear and not able to be misinterpreted.

Imagine, for a moment, that you are in a cellar, with a walkie-talkie, directing a friend in the street. Your friend is pretty stupid, and will only do exactly what you tell him. He will also ALWAYS do exactly what you tell him. The problem - to get him across a fairly quiet street, without being run over.

"Turn so that you are facing the road"

"O.K."

"Move forward to the edge of the pavement"

"OUCH!!!" (Someone just walked into a stobie-pole)

"If there aren't any obstacles in your path, move forward to the edge of the pavement. Otherwise walk a couple of paces up the pavement and try again"

"O.K."

"Are you on the edge of the pavement?"

"yes"

"Look left, then right, then left again. If you cannot see any cars, cross the road. If you can see a car, repeat this instruction. Continue repeating it until you can cross".

At this point your friend will either walk in front of a semi-trailer, a refuse cart or steam-roller (which you failed to warn him about), OR he will spend the next week looking left and right, because there is a stationary car right alongside him. And as that is YOUR car and you are waiting for him to cross the road - all is lost.

That is how it is when you try to communicate with your '80. You must learn how your DUMB friend thinks, and then you can think like it does. With practice, you will learn to win most of the time.

## \*\* THE FOUR OPERATING MODES \*\*

The '80 can have up to four separate operating modes. These are:

### \*\* COMMAND MODE (called the CALCULATOR MODE in Level I)

You are in the COMMAND MODE whenever the last line printed on the screen (which I will refer to as the V.D.U. from now on) reads... ">"

(On the System '80, the first ">" after power-up is not what it appears to be, and you should hit NEW LINE to enter COMMAND MODE)

The ">" symbol is called the "PROMPT" and is the '80's way of telling you that it is waiting for an instruction.

### \*\* EXECUTE MODE

This mode is entered whenever the '80 is actually following an instruction, which may have been stored in memory, or input while in the COMMAND MODE.

### \*\* EDIT MODE (not available in Level I)

This powerful mode (one of the major reasons why I finally bought an '80) allows you to edit program lines efficiently. This mode will be the subject of a later article.

### \*\* MONITOR MODE (not an inbuilt feature of Level I)

Accessed by typing "SYSTEM" and then pressing ENTER (LINE FEED on the SYSTEM '80). This produces a different PROMPT-sign, "\* ? ", and will be discussed separately in a later article.

This month we will examine the Command Mode, and do some...

## \*\* SIMPLE MATH IN COMMAND/CALCULATOR MODE \*\*

The four basic math functions are simply represented by...

ADD	+
SUBTRACT	-
MULTIPLY	*
DIVIDE	/

We must also remember that if the '80 is to output anything at all to the V.D.U., it has to be told to PRINT. Typing 8+7 and pressing the ENTER or NEW LINE key, will not produce the result you want. But...

PRINT 8+7	will produce...
15	followed by...
READY	says "finished..."
>-	says "ready and waiting..."

Exactly the same techniques can be used for other mathematical operations.

## \*\* INVISIBLE BRACKETS \*\*

The problem is simple enough: Ten oranges at 20 cents, plus 12 apples at 11 cents, plus 6 eggs at 18 cents. How much?

Now, we could work out 3 separate multiplication sums, and then one addition - and the answer would be correct. But that is inefficient. So, groping for our schoolday algebra, we come up with (10 oranges at 20c) plus (12 apples at 11c) plus (6 eggs at 18c) equals ?

Try... PRINT(10\*20)+(12\*11)+(6\*18)

now try... PRINT10\*20+12\*11+6\*18

amazing isn't it? Your "dumb" '80 put those brackets in for itself - or did it?

A new problem...13 gizmos at 95 cents plus 10% sales tax, plus 11 floggle-toggles at 12 cents plus 17% sales tax...easy enough?

Try... PRINT(13\*95\*1.1)+(11\*12\*1.17)

now try... PRINT13\*95\*1.1+11\*12\*1.17

Yes again, "dumb" old '80 put the brackets in - or did it?

So this time we have... 11 thingummies at 42 cents plus 11 gizmos to suit the thingummies at 4 cents.

PRINT (11\*42)+(11\*4) works, (though inefficiently) as does

PRINT 11*42+11*4	BUT, when you try
PRINT 11*42+4	??????????
PRINT 42+4*11	??????????

What happened? Well, the invisible brackets were there, and the problem worked like this...

$(11*42)+4$

and this...  $42+(4*11)$

while what you wanted was  $11*(42+4)$ . Why?

Well, there has to be a set of rules somewhere, and there is one for math. The '80 actually looks at a math problem several times, working out different bits and pieces each time. The rule says that 1. IF THERE ARE MULTIPLE BRACKETS, WORK OUT THE MIDDLE ONE FIRST, THEN THE NEXT ONE OUTSIDE, AND SO ON...

Then, within each bracket, or taking the statement as a whole if there are no brackets or if all the bracketed parts have been worked out already... IN ORDER

2. expotentialiation e.g.  $12 \uparrow 4$
3. negations e.g.  $-X$
4. \* and / (left to right)
5. + and - (left to right)
6.  $< > = < = > = < >$  (left to right)
7. NOT
8. AND
9. OR

Don't worry that you maybe don't know the meaning of all these long words like "expotentialiation", because we will discuss all of these before too long. What you should worry about is this...

Rule 1 says that the '80 will work out the middle bracket first, and then work on out toward the outsides of the problem. Now this doesn't matter too much in the tiny shopping list problems that we just did, but later on, when you get to be writing more massive problems (where the same equation needs to be run several hundred times) the extra time needed will start to become important.

With practice, you will learn where a bracket is NECESSARY, and where you can get away without using it, because Rules 2 to 9 determine the order of working out the various parts of the problem.

\*\*\*\*\*

\* G.T. BASIC RULE ONE - ONLY USE A BRACKET WHEN YOU MUST \*

\*\*\*\*\*

Now, some of you "old hands" out there are probably thinking that this seems a bit unimportant to qualify as Rule one. So you may like to time the following two COMMANDS, and see for yourselves.

TEST ONE

>FORK=0TO300000: Z= (1+ (2+ (3+ (4+ (5+6) +7) +8) +9) +10) : NEXT: PRINTZ

TEST TWO

>FORK=0TO300000: Z= 1+2+3+4+5+6+7+8+9+10: NEXT: PRINTZ

LEVEL I users, can substitute...

>F.K=0TO300000: Z= etc. etc. :N.K:P.Z

The time saving is better than 15%!

\*\* NUMERIC VARIABLES \*\*

If you've tried those tests, you've just made another discovery, because the '80 will have printed "55" before it came back with READY and the PROMPT. But you didn't tell it to print "55"! The last part of the instruction was PRINT Z, so where did the "55" come from?

Back to your algebra again.  $Z=1+2+3+4+5+6+7+8+9+10$  tells the '80 to store the answer in something called "Z". Then, later, you told it to PRINT Z and it printed the value stored there instead of printing "Z".

That something "Z" is called a NUMERIC VARIABLE. You have a lot of numeric variables - all with different names...How many you have will depend on whether you are using Level I or Level II. In Level I you have 26 NUMERIC VARIABLES, called A, B, C...right on up to Z, which can all be used to store numbers. You also have something called an ARRAY, which will be the subject of another article, very soon.

In Level II you have...better sit down for this...3066 VARIABLES just for storing numbers! But for the time being we'll just worry about the 1022 easy ones. 26 of these are the same as in Level I, A through to Z. Then there are another 996, starting with AA going right through AZ and on through AO to A9, before starting again with BA, and so on to Z9. Level II also has arrays - they're extra and again, will be discussed properly at the appropriate time.

To put a number into a NUMERIC VARIABLE, the regular BASIC requires that you use the command LET.

(continued on P. 8)

**\*\* BETTER BYTES \*\*** a wander through a "bit" of a jungle conducted by Peter Hartley.

(Readers' contributions are invited for this column)

First of all, I must apologise for those readers who have recently submitted software to this magazine. While I'm doing my best to find the time to get through it all, I must again ask for your patience and indulgence. Please understand that MICRO-80 is a wholly part-time operation and all of us have to spend time on mundane things like earning a living. Plans have now been made for a complete reorganisation of this area of our operations which, when effected within the next couple of weeks, should cure these problems once and for all. As a result of these changes, this will be my last edition of MICRO-80 as Software Editor - although you'll be stuck with my software selections for September! I'll still be doing a goodly chunk of columnizing, and those regular features which suffer from my association will continue unchanged.

**\* GOD LIVES AT FORTH WORTH, TEXAS \***

An open letter to Tandy (Australia)

Gentlemen,

Does anyone in your organisation possess a customised motor-car? Something fitted with four-barrel Holly's, polished ports and a shaved head? And does it ever need service?

Mechanics' attitudes to such a customised vehicle may range from bewildered amusement, through amazement, to pure green envy. But the owner WILL be able to get that vehicle fixed, and when he comes to collect it he will not find those treasured "extras" and "mods" all packed up in a polythene bag and thrown in the boot, together with a note saying...

"REMOVED JUNK - RETURNED TO ORIGINAL..."

So how come that my CPU (serial No. 19500) had everything ripped off and the above message duly inscribed on docket No. 071547?

What gives the Gods of Fort Worth and Rydalmere the ultimate right of deciding what modifications I may or may not run on MY TRS-80?

It took you a couple of years too long to produce SCRIPSIT. I and many others use another word processor. One that demands modifications. TRS-80 owners who use Electric Pencil, or who modify their TRS-80 to run Johnnie's train set, are still YOUR customers, entitled to expect reasonable service and not to have their modified TRS-80's violated and, most certainly, not to receive high-handed crap like the above message.

When I telephoned to (a) enquire and (b) complain, I received polite, firm, courteous disinterest. Everything that I heard carried the subtle implication 'hat Tandy believe that customers exist purely for Tandy's benefit'. The customer, it seems, is only right when he agrees that the TRS-80 is perfect only when a virgin! Anyway, the technicians work from documentation that relates to the standard unit!

If Tandy technicians cannot understand commonly applied modifications then why not think of the customer's interests and get your technicians trained in the vagaries of the most popular mods? However bloody awful a customer's mod may look to the Tandy purist, it is still the customer's mod, and the customer wants it. Haha - not in a plastic bag.

Now we come to the rest of your technician's delightful little message...

"...RECTIFIED INTRODUCED FAULTS"

That is about as subtle as a Centurion Tank at a Girl Guides' Picnic.

Please will you explain why, before repairs, my modified CPU would work with any one of the expansion interfaces that it was mated with - except my own! If MY modifications were stopping it from working properly why was it necessary to replace 3 \* 741567's? Now that all my modifications have been replaced WHY DOES IT STILL WORK?

Also, in view of the claims you've made in overseas advertising for the speed of your worldwide 'service', how about letting me know why, when my equipment was taken to your Glen Wondwong store (8108) on July 30th, it was not received by Computer Quality Control until August 11th? Why was I advised on August 8th that the equipment was lost? Why do customers lose their importance on Friday afternoons? How come that my system was apparently re-dispatched from your warehouse to another Tandy store without paperwork, instead of being sent to Computer Quality Control? Since it was dispatched from Computer Quality Control on August 13th, why didn't I get the unit back until August 18th?

Why are repairs left sitting in Sydney until the next rotational shipment to the originating Tandy store even when the customer has indicated that he is willing to pay for air-freight? Why has this happened to me twice? (Last October it sat in Sydney after repair for FIVE days!)



Why do you treat your customers this way?

Finally, about your utterly disgraceful attempts to minimise warranty claims by putting pathetic 'seals' all over the place. MY TRS-80 is MINE. If I want to open it up and stare at all the magic plastic boxes with shiny metal legs all around their edges - then that's my right. If the unit fails within a reasonable period of time, the customer has the right (legally enforceable in most States) to have it fixed unless there is evidence of improper handling. Breaking a "seal" is NOT evidence of improper handling, and I don't believe that plugging sensibly priced RAM could be construed as improper handling either.

Why do you treat your customers this way?

When you've created a product that scores of thousands of people all over the world love, why do you treat them so badly that they develop a hatred for their best friend's creator? Give the customer some support!

Peter Hartley

Unhappy Customer.

**\*\* SEALS AGAIN!"**

You can make a good imitation of the TANDY seal, with a mixture of AIRFIX polystyrene cement, cherry red nail polish and AIRFIX orange model paint!

**\* TIP OF THE MONTH** for situations requiring password protection to determine access levels...

Open a file, as OPEN. "O".1. "PASSWORD.XYZZY" or any other password you may care to name, and then CLOSE.

In your main program, use the following... (For Disk BASIC only-Ed.)

```
10 CLS
20 INPUT "PASSWORD":D$
30 ON ERROR GOTO 70
40 D$="PASSWORD."+D$
50 OPEN "I".1.D$
60 GOTO NNNN1
70 CLS
80 PRINT "PASSWORD REJECTED"
90 PRINT "RESTRICTED ACCESS ONLY"
100 RESUME NNNN2
NNN1 CLOSE...
NNN2 CLOSE...
```

**\*\* SUPERDOS - by Peter Hartley.**

Finally got my podgie pinkies wrapped around the long awaited NEWDOS 80 last week and started a whole new learning experience. The manual is very detailed - about 200 pages - well indexed, and needs careful study because NEWDOS 80 is a total rewrite and not much related to NEWDOS+, let alone TRSDOS. Although I'm still learning how to drive this package, I can say that it probably represents the single greatest advance in hobby and small-business computing since the TRS-80. It is immensely powerful, giving one many of the features normally found only on our bigger mainframe brothers.

Doubtless I'll be raving on at more length in later issues, but, just to whet your appetite...

Random files without having to use FIELD, LSET, RSET, CUT, CUS, CUD, MKI, MKS, MKD... No more worrying about 255 byte sectors...Write up to 4K to disk in a single PUT...Mix or match 35, 40, 80 96 or what-have-you track drives - and the system will remember what is where...One of your disk drives a bit unreliable? Change the number of times that the disk controller will "look again" before demanding that error message...

- 00000 -

Continued from P. 6 (G.T. BASIC)

E.G. LET Z=55. The BASIC used in the '80, however, does not make this demand, and a simple Z=55 will put the value in. PRINT Z, will get it back and print it on the V.D.U. But the value 55 is still stored in Z. To change it, you have to put a new value in...Z=77 for example.

Since the '80 knows what Z is worth, it only seems reasonable that it would understand instructions like Z=Z\*1, Z=Z\*2, Z=Z+77+Z\*2 and so on and, of course, it does. If you're a beginner, why not try a few of these for yourself, following them with PRINT Z (just to satisfy yourself that they work)?

Next month we will look at one of the power-house instructions of BASIC - the amazing FOR / NEXT LOOP, and run some tests to see how good the '80 is at remembering things in the COMMAND MODE. We will also show some of the old hands how they can save a whole mess of run-time while using FOR / NEXT LOOPS, and introduce G.T. BASIC RULE TWO.



Boris's play however was seriously flawed. His ability at corresponding response times appeared to be comparable to Chess Challenger and, in fact, they had drawn a match of 14 games, but there was a certain rigidity about Boris's style of play which made him predictable and invited stereotyped attacking strategies. It is essential for any program which cannot be modified or updated, as Boris was then, to have varied treatments of similar positions, if it is not to be tired of. Boris is now equipped with a plug-in module facility which enables up-dates of the program to be inserted into the basic memory as improvements are made, which is an excellent idea.

A month after these initial comparisons were made, Chess Challenger improved their model 10 and added a voice program to rename it the 'Voice Challenger'. This was an improvement all round with stronger play on every level, a greatly expanded book of opening repertoires and provision for selecting the opening against which to play. The one hour and 24 hour levels had disappeared in favour of an infinite level which could be halted at any time to play 'best so far', or allowed to continue until search exhausted - up to several days.

The development program for Boris meanwhile was taken over by the authors of Sargon and all comments regarding Sargon from hereon can be taken to include Boris with its additional facility for updating so that Boris is now the machine for Sargon-squashers who neither have a mini-computer nor are interested in ping-pong.

The authors of Sargon have now produced four programs with inbuilt obsolescence rendering each preceding program out-of-date about every six months. It is a tribute to their zeal and dedication that this formerly amateur husband and wife team have been able to effect substantial improvement in a short space of time with each successive Sargon and stay ahead of the leap-frogging competition.

The Sargon 2.5 probably represents the state-of-the-art in cassette programs for mini-computers and I believe it has just become available in Australia. It is claimed that this program has a playing strength of 1600 (good A Reserve Adelaide Club strength) at the three minute level. This rate of play has become the yardstick by which computer play is judged and computer tournaments conducted, and it seems valid that if computers are to be taken seriously in chess, they must then be assessed on the basis of their results within a reasonable time frame. Hence the battle for chess-computer supremacy has led to initiatives on two separate fronts undertaken by different schools of thought mobilising different hardware. Programmers of the mainframe computers have aimed to maintain the advantage they hold through sheer speed of operation by increasing still further the number of positions evaluated per second, and hence the depth of search or number of plys. Since the number of positions looked at increases astronomically with each successive ply, one machine would need to perform enormously faster than another in a given time span to search an extra ply. The mainframe specialists claim a deeper search by several plys over the micros, giving them an unassailable advantage. They feel it is simply a question of horsepower.

However, their rivals in the field of micro computers point out that this is not the way the chess master reasons and depth of search is useless without correct evaluation of positional advantage. The Spracklens with Sargon and other of their competitors are ingeniously devising ways of eliminating wasteful searching by cueing the program with correct thematic responses to thousands of typical middle-game and end-game board positions. This enables positional-judgment as developed by chess masters based on recognition of familiar positions and pruning of non-productive lines of play from the search. This must be the correct approach if computers are ever to compete with masters (they cannot, yet).

This approach is starting to yield impressive results and Sargon 3.0 (not yet marketed) gave strong competition to mainframe entries in this year's North American Chess Championship almost beating second placegetter Belle. Undoubtedly, the mainframe programmers must pursue this line of research if they are to maintain their lead over the micro-processors, and it has been adopted already to some extent by Chess 4.9 the World Computer Chess Champion.

There is no doubt in my mind that Sargon 2.5 is already a worthy opponent for the majority of serious players of club strength and a program which would probably elude defeat by a talented beginner at the game for at least a year, and possibly longer, if the present rate of progress is sustained.

Computers do not yet figure as serious contenders in master play as I have mentioned, but Chess 4.9 is currently rated at 2100 which would enable it to contend for the South Australian Championship (but not win it). It should be realised that there is still a vast gap between this level of expert play and international master and grandmaster strength. There are about 150 grandmasters in the world. When one considers the number of Ph.D's produced by any large university in a year, the merit of this title will be realised. One must also appreciate the significance and prestige of chess in many other parts of the world. There are no Australian chess grandmasters. In the Soviet Union, where chess masters are accorded a high status and income, there are over 4 million registered players. (continued on P. 40)

# ASSEMBLY LANGUAGE PROGRAMMING by E.R. PAAY

The previous parts of this series have discussed most of the commands and instructions available to the machine language programmer and some examples have been given on how to use these instructions. By now the reader should have enough knowledge to write machine language programs. This part therefore shall contain a general discussion and several commonly used subroutines.

## THE EDITOR ASSEMBLER.

At this stage it might be useful to mention the role of the editor assembler as we get numerous requests from readers asking what exactly an editor assembler does. The editor part of the assembler allows the user to type in the source code and edit it in very much the same manner as BASIC allows us to edit lines etc. After typing in the source (or loading it from tape) the assembler then allows the user to "assemble" the source code into the object code or machine language, which is the actual program which can be executed by the CPU. Apart from the functions mentioned it will also save both source code and object code on tape or disk whatever the case may be. There are several different editor assembler programs available for the TRS-80 user. First there is the Tandy version which is written by Micro-soft; this is a good assembler which is relatively easy to use, however, it has been superseded with a new version called editor assembler plus. This program has all the features of the old version but it allows the use of MACRO'S and it can assemble straight into memory which saves tedious tape dumps while debugging. Included with this editor assembler package is also a very powerful debugger program which allows single stepping through programs etc. This means that this assembler is much better than the old Tandy version and I would therefore recommend it for the tape bound TRS-80 user (or SYSTEM-80 user). For the disk user there is the disk assembler by Micro-soft. This is a MACRO assembler also but is somewhat tedious to use as the source has to be written using an editor called EDIT80. When that has been done, you have to exit the editor which will then dump the source to disk. After that has been done, the assembler has to be called up which is a separate program. This will then create yet another file on disk. When it is finished the linking loader has to be called up which is a separate program again, to create the final program which can be run. This is a lot of work but, at the moment, it is the only readily available disk assembler with MACRO capabilities that I know of. It might be worth mentioning that a disk editor assembler is included with all NEWDOS + disk operating packages. However, this is the same as the Tandy tape version with disk capabilities added. I myself use the NEWDOS version for short simple programs and the Micro-soft disk assembler for long complicated programs.

## MACRO'S.

Some readers probably don't know what a MACRO is so a short description shall follow. A MACRO automatically generates assembly language instructions using user supplied parameters. A MACRO is in some respects similar to a subroutine except that whenever a MACRO name is placed in the program the source code associated with that MACRO is placed in line with the program at the location where the MACRO name is placed. For example:

```
100 SCAN    MACRO  ADDRS
110         LD     A,(ADDRS)
120         OR     A
130         JR     NZ,DECODE
140         ENDM
```

Line 100 tells the assembler that a MACRO named SCAN is being defined ADDRS and is a dummy label used by the macro. Line 130 tells the assembler that this is the end of the MACRO. This means that the code between line 100 and line 130 is going to be the actual MACRO definition. Of course many more lines and dummy labels can be used in an actual MACRO. The purpose of the example above is to scan the keyboard. Now if the above MACRO is used in a program it might look like this:

```
200         SCAN    3801H        ;scan row one.
300         SCAN    3802H        ;scan row two.
400         SCAN    3804H        ;scan row three.
```

After the program is assembled the MACRO'S will have been expanded and it will look like this:

```
0200         SCAN    3801H
45E2 3A 01 38         LD     A,(3801H)
45E5 B7              OR     A
45E6 20 0F           JR     NZ,DECODE
0300         SCAN    3802H
```



```

45E8 3A 02 38      LD      A,(3802H)
45EB B7            OR      A
45EC 20 09            JR      NZ,DECODE
                   0400    SCAN      3804H
45EE 3A 04 38      LD      A,(3804H)
45F1 B7            OR      A
45F2 20 03            JR      NZ,DECODE

```

It can be seen how only three lines were expanded to 9 lines, and the dummy label was replaced with the actual parameter following the MACRO name. Of course the macro was kept short for the actual example. Normally a MACRO might be much longer using a range of parameters.

#### PROGRAM PREPARATION.

Preparation before writing a machine language program is very important, much more so than with BASIC. Many amateur BASIC programmers will type in their programs without drawing flowcharts by typing it in a few lines at a time and trying it as they go along. Although you will never end up with a good program this way, it can be done. Using this approach with assembly language will usually end up in disaster. For good quality assembly language programs the following procedure is recommended. Before writing the program, make sure in your mind WHAT has to be done and outline the rough procedure on paper. Then when you are SURE as to the approach you are going to take, draw up the flowchart. Draw a general flowchart first, naming each block and expanding each block into smaller flowcharts if necessary. Initially keep your mind on the overall program flow - don't get bogged down with details at the start - handle these later when the main blocks of the flowchart are expanded into their own flowcharts. With some experience, you will find this will make programming much easier both in BASIC and in assembly language. When and only then start writing your program. Avoid using immediate jumps, give your main blocks of your program meaningful labels and CALL them, structure your programs as much as possible - don't make a spaghetti bowl program with jumps back and forth right through the program. Also, it is very important to include comments in your program because you WILL FORGET what you have done when you are debugging or when you want to alter something some time later. And as I said before, make your labels meaningful so you know what they are without having to look them up.

#### SUBROUTINES.

In this section several commonly needed routines are listed which can be used by the reader as subroutines to make life somewhat easier. First, we have a HEX to ASCII decimal conversion routine, a 16 bit value is passed in the HL register pair, DE should point to the last byte of a 5 byte buffer into which a 5 digit ASCII decimal result will be placed. All registers are used; when it returns DE will point to the start of the buffer. No line numbers are given. They will have to be added by the user.

```

HEXASC  LD      C,5
        LD      B,10
LP10    PUSH    BC
        XOR     A
        LD      C,11H
        JR      STARTH
LP11    SUB     B
        JP      P,NREST
        ADD     A,B
STARTH  ADD     HL,HL
        RLA
        JR      CONT
NREST   ADD     HL,HL
        RLA
        INC     HL
CONT    DEC     C
        JR      NZ,LP11
        RRA
        POP     BC
        ADD     A,30H
        LD      DE,A

```

```

DEC      DE
DEC      C
JR       NZ,LP10
RET

```

The next program does the opposite. It converts a string of ASCII decimal digits into a hex value and places the result in the DE register pair. On entry IX must point to the start of the ASCII string; all registers are saved except HL and DE.

```

ASCHEX   PUSH      DE
          LD        DE,0
LOOP     LD        A,(IX+0)
          SUB       30H
          ADD       A,E
          JR        NC,ASCO
          INC       D
ASCO     LD        E,A
          POP       BC
          DEC       B
          RET       Z
          PUSH      BC
          EX        DE,HL
          ADD       HL,HL
          PUSH      HL
          ADD       HL,HL
          ADD       HL,HL
          POP       BC
          ADD       HL,BC
          EX        DE,HL
          INC       IX
          JR        LOOP

```

The next routine will convert HEX to ASCII HEX so that it can be displayed on the display. The A register contains the HEX byte and the DE register pair will contain the two ASCII characters on return. All other registers are saved.

```

ASCCON   PUSH      AF
          AND       OF0H
          RRCA
          RRCA
          RRCA
          RRCA
          CALL      CONV
          LD        D,A
          POP       AF
          AND       OFH
          CALL      CONV
          LD        E,A
          RET
CONV     CP        GAH
          JR        C,CNT10
          ADD       A,7
CNT10    ADD       A,30H
          RET

```

Finally we have a routine to convert ASCII HEX to binary HEX. DE should contain the two ASCII characters; the result will be returned in the A register. All registers are saved except DE and A.

```

HEXCON   LD        A,D
          CALL      CONVRT
          RLCA
          RLCA

```

Continued on P. 18

## \*\*\*\*\* ESF FORUM by Charlie Bartlett \*\*\*\*\*

Welcome to the first ESF FORUM. Over the next few months we will delve into the mysteries of the ESF, and anyone that has not bought one yet will notice by reading these columns that the ESF is not "JUST A FAST CASSETTE DECK".

## \*\* PROGRAM CHAINING \*\*

One of the features of the ESF is its ability to chain BASIC programs together and pass variable values from program to program. This can be very useful for instance if you have a 32K program (or larger) and a 16K machine, because you can break the program down into two (or more) modules and then have your program load and run them separately while passing the values from module to module. This is called overlaying. The only thing to consider when chaining is that for the variable values to remain intact they must stay clear of the overlayed BASIC programs. This can be done in one of two ways. Either the first program must be longer than the second or the variable pointer must be poked into high memory out of the way. The first way is easy and doesn't need further explanation, but for the second way, you need to know what to POKE and where. We will demonstrate this using the program "HOUSEHOLD ACCOUNTS", which was on the free software tape or is listed in its original form in the May edition. If we are going to chain a program we may as well go for broke, so we will break this program down into ELEVEN modules. It could, of course, be done in less, but this is more of a lesson in chaining, and besides which, by making the modules as small as we can, it is possible to increase the size of the buffer.

Also with the DATA I/O program loaded in there is not enough room left in memory for this program to run in one piece anyway. Examine the program listing for module one...

5 POKE16634,74

This is the statement that pokes the variable pointer up into high memory. The formula for arriving at the desired value is..

The value present with your FIRST module loaded plus the number of K of your LARGEST module times 4. So if your largest module was 4K then the statement in your program would appear thus...

5 POKE16634,PEEK(16634)+16

One other thing to remember when chaining programs that are to pass variable values between each other is that if there are any CLEAR or DIMension statements in the first module then this module must be loaded first and then not loaded again, which is why the POKE, CLEAR and DIM statements in this program are in module one, and if you examine the listing you will see that the program never returns to this point.

Program line number 15 is a POKE to disable the break key except when the ESF is running. This is to prevent accidental interruption to the program while live data is present. It is a good idea to incorporate this into any program when you don't want to lose data.

15 POKE16396,23

To enable the break key when you have finished with the program, without having to switch off, POKE16396,201

When all the modules are loaded onto a wafer, (it will need to be a 20 ft. wafer), don't remove the reflective sticker as you would normally do as the subroutine in the @SAVE and @LOAD modules uses this to sense if the program wafer is inserted or not. Leave it to the program to tell you when to insert or remove wafers. To start the program, FIRST load the DATA I/O wafer. Then insert the Household Accounts wafer and type @LOAD1. The ESF will first of all load in module 1 and then with no action on your part, will load in module 2, which is the MENU. The program will then load in the modules as you select them from the menu...It's as simple as that. See program listing on p. 36

## \*\*\*\*\* SPECIAL OFFER - EXATRON STRINGY FLOPPY \$352.50 incl. p&amp;p \*\*\*\*\*

All Exatron Stringy Floppies sold by MICRO-80 will include the special chained version of HOUSEHOLD ACCOUNTS, developed by Charlie Bartlett. When used on the ESF, this program is powerful enough to perform many of the accounting functions in a small business. Remember, the ESF comes complete with a comprehensive manual, a 2 way bus-extender cable, its own power supply and 10 wafers of mixed length. One wafer contains the Data Input/Output program and another the HOUSEHOLD ACCOUNTS program.

CAN'T MAKE UP YOUR MIND ABOUT THE ESF?

Then send in \$5.00 for a copy of the manual. We will refund your \$5.00 IN FULL when you purchase an ESF.

LETTERS TO THE EDITOR

From: L.R. McDonald, Qld.

Could you please make some comment in your magazine with regards to the suitability of Dick Smith disk drives used with the TRS-80 as I have heard that some people have had considerable trouble with them not working properly with the TRS-80. Does NEWDOS 80 get around this problem? Do the programs that you supply on disk only suit the Tandy disk drives.

(We too have heard of some people having trouble with PERTEK (i.e. Dick Smith) disk drives in TRS-80's. However, if set up correctly, they should work without any trouble. We should know, between us, the Editorial staff are running 6 of these drives and we have mixed them successfully with a Shugart drive which was the original brand supplied by Tandy. The clue to setting up PERTEK drives is in section 3.5.3 on page 18 of the manual supplied with each drive. Use the following procedure:

- 1) Turn the drive upside down with the rear (open end) of the drive towards you. Unscrew the four screws holding the aluminium bottom plate and remove it.
- 2) This will reveal the component side of a printed circuit board. Towards the bottom right hand corner of the board you will find a 4 way DIP switch (S1). It is generally bright red with white rockers and stands out from the other components. You must set this switch to correspond with the drive number you wish this drive to be, i.e. if it is to be drive 0 then switch 1 must be closed and all the others open. Section 3.5.3 tells you the settings to use
- 3) Now look along the row of integrated circuits which runs up the centre of the PC board. Between the second IC from the bottom (U8) and the short IC to its left (U6), there are two plated-through holes in the PC board, very close together, with a very fine track joining them. This is track DC mentioned in Section 3.5.3. Take a sharp razor knife and cut this track. Don't overdo it and cut through the PC board but do make sure that the track is completely severed and that there are no slivers of metal still making contact across it. (Cutting this track allows the switch (S1) to do its thing).
- 4) Now, lying on a diagonal line between U8 and S1, is the resistor pack U2. It is generally a bright blue and stands out from the I.C's around it. This pack should only be left in place in the highest numbered drive on the cable, i.e. if you have three drives connected (0, 1 and 2) then the resistor pack should be left in drive 2 and removed from drives 0 and 1. If you are running only one drive then it must be set up as drive 0 and the resistor pack should be left IN.

If you follow the above procedure, then you should have no problems in using PERTEK disk drives on a TRS-80. Incidentally, most disk drives need to be set up in the same way on a TRS-80.

As far as NEWDOS 80 is concerned, there is no problem left for it to solve. What it will do, however, is enable you to use a PERTEK drive to its full 40 track (100K byte) capacity. And, what is more, you could use it as a 40 track drive on the same cable as a Tandy 35 track drive or a Micropolis 77 track drive or both!

All the programs supplied on disk by us will run on a standard 35 track drive. This means that they will also run on a 40 track drive. - Ed)

From: C. Stobert. Victoria.

Our System 80 was purchased between receipt of your May and June issues. Consequently, the differences in tone between your unequivocal praise in the May issue and the product review "rip-in" in June, is disturbing. I had hoped for something better having just been stung by Tandy for over \$26 for repairs only six weeks out of warranty.

The System 80 specification quotes a video output of 2V p/p whilst the Tandy spec. states 1.4V. I would like to use my System 80 with a Tandy monitor but am concerned that the use of an input voltage 50% higher than specified might affect the monitor.

We are not able to run "Super Mastermind" from your software cassette (Mr. Stobert also has a TRS-80 Level I 4K machine - Ed). After loading the PRINT MEN produces -239 and LIST shows the program to run out somewhere in the middle of the instructions.

(We do seem to have ruffled a few feathers over the June review of the System 80, not least in the Dick Smith camp. First, to deal with the apparent about-face. At the time of producing the May



issue we had not sighted a System 80 and that was clear from our comments. We congratulated the Dick Smith organisation for persevering with the project over such a long time and applauded the extra competition it has introduced to the Australian market place. We still feel that way. Dick Smith Electronics played a significant part in bringing the System 80 to fruition and deserves to be applauded for it. By the June issue we had our own System 80 and had had time to examine it in detail and put it through its paces. If you reread the June review, as I have just done, I think you will find that it is, overall, positive about the System 80. It certainly expresses disappointment that a new system which had been widely advertised to be compatible with the TRS-80 turned out not to be in some significant aspects (missing keys, different 32 character format, different printer addressing). Certainly, we criticised the internal cassette deck for its poor performance and everything we have heard since has confirmed that view. We now have a significant number of subscribers who own System 80's. At least 70% of those who have written, phoned or visited us, are having loading problems with their cassette decks. For our part, we are recording all our tapes at higher levels to cater for the System 80. No small matters these, particularly to suppliers such as ourselves. The lack of compatibility has cost us time and money. But that review also said things such as (p.10) "Now, even that sum puts the Dick Smith System 80 in a very, very favourable light". and page 9 "The rest of the internal construction was excellent.....the System 80 appeared to be a solidly constructed piece of equipment" etc. Incidentally, our review was inaccurate in one respect; we said that only the Z80 was in a socket. In fact, the Level II ROM's and the memory chips are also in sockets - sorry about that, it must have been the late hour when we examined the internals.

I might say that a similar review of the TRS-80 written before the XR3111 modification was developed would have been equally as critical and there is certainly evidence of penny pinching in that machine too. Solder-coated edge contacts on the CPU board (the System 80 has gold-plated contacts) - no data separator in the floppy disk controller etc. So there you have it, the perfect micro-computer is yet to be built!

As far as the different video output levels are concerned, the System 80 will not harm the TRS-80 monitor. Its output voltage is well within the tolerance of the monitor. We have been running a TRS-80 monitor with our System 80 for some months now and no problems.

Super Mastermind belongs in our "Red Faces" department. When we first published it in the December issue, we explained that several lines needed to be deleted to run it in a 4K machine, then we made the master tape with those lines in. Sorry about that. If any reader would like the 4K version, please return your free cassette and we will replace it. Needless to say, we have now reworked the master cassette. - Ed)

#### SYSTEM 80 - 32 CHARACTERS PER LINE UNDER SOFTWARE CONTROL

On the TRS-80, double sized characters (same height, double length) on the CRT display are selected by the basic commands

1. PRINT CHR\$(23)
- and 2. OUT 255, 8

The PRINT CHR\$(23) command when executed, sets a flag in location 16445 to indicate that the user has set the 32 character per line format. If the OUT 255, 8 instruction is executed, then the flag in location 16,445 will not be set. Each time a character is to be printed to the screen, the status of location 16,445 is checked. If it is 0, then the 64 character/line format is set, and if it is 8 then the 32 character/line format is set.

TRS-80 owners may like to try the following short programs:

- A.     10     OUT 255,8  
       20     END.
- B.     10     OUT 255,8  
       15     FOR A= 0 to 2000: NEXT A  
       20     END
- C.     10     POKE 16 445,8  
       20     END
- D.     10     POKE 16445, 0  
       20     END

No effect will be seen on executing program A as it executes very fast and on returning to command mode the computer will check the status of location 16445 before printing "READY". As the "OUT" instruction will not affect this location, the screen format is reset to 64 characters/line. Program B introduces a delay so the 32 character/line format can be viewed for approximately five seconds before reverting to the 64 character/line format.

Programs C and D manipulate directly the format flag so that when the next character is printed on the screen the format is set accordingly.

The System '80 differs quite markedly from the TRS-80 in this area. Firstly, the hardware associated with port address 255 (FFH) bit line 3, has been deleted. In its place is a hard wired switch called "VIDEO CUT". When activated, double-sized characters appear, and may be disabled by releasing the switch. No software control of this facility is provided.

Executing the basic command PRINT CHR\$(23) will cause the System '80 to print strings with a space inserted between each character from the next print statement in the program. Another way of doing this is by executing the command "POKE 16445, 8", which sets the format flag.

The display reverts to the normal single spacing format for print statements when the basic command CLS OR POKE 16445,0 is executed.

The hardware modification described later in this article will allow the following modes of operation when the basic statement is executed.

1. POKE 16445,8 selects double spacing mode. Each print statement executed after this command will have a space inserted between each character. Operates in both 32 and 64 character per line format.
2. POKE 16445,0 de-activates double spacing mode.
3. OUT 255, 8 selects 32 character/line format.
4. OUT 255,0 selects 64 character/line format.
5. PRINT CHR\$(23) is equivalent to the combination OUT 255,8 : POKE 16445,8
6. CLS is equivalent to the combination OUT 255,0 : POKE 16445,0.

N.B. \* The video cut switch when activated will select the 32 character/line format and override any software selected format.

\*\* On returning to basic command mode (i.e. after a program end statement is encountered) the 64 character/line format will be selected if the flag in location 16445 is not set at 8. If this flag is set at 8 then the double spacing mode is automatically set and hence the greeting R E A D Y.

#### MODIFICATION PROCEDURE

You will require

- A Philips head screw driver
- A low power ( 40W) soldering iron
- Some fine multicore solder
- A diode 1N 914, 1N 4148 or equivalent
- Some plastic-coated flexible wire.

1. Switch off the System '80 and remove the plug from the power point.
2. Remove the System '80 moulded top cover by undoing the screws located on the bottom of the unit. Very, very carefully unscrew the keyboard unit and shift the upper right-hand side to expose the circuit board located between the CPU board and the cassette unit. This board has the connector for the video output, the second cassette drive and the video cut switch.
3. Install the two links and the diode as shown in the diagram. The diode must be installed with the cathode (indicated by a black or a white bar at one end of the device) connected to the 74LS175 IC pin 6. Check that no solder dags are shorting the IC pins before powering on.
4. Type in the following basic program:
 

```
10 PRINT CHR$(23)
20 END
```

When run, all characters on the screen should be double-sized and READY should be printed double spaced.

5. If it does not work, check your wiring, and ask someone who knows about electronic circuitry to look it over.

Now replace the keyboard and cover and check out the following demonstration program:

```

10 PRINT CHR$(23)
20 PRINT "THIS IS PRINT CHR$(23)"
30 GOSUB 130
40 POKE 16445,0
50 PRINT "THIS IS NOT DOUBLE SPACED AFTER POKING ZERO"
55 GOSUB 130
60 OUT 255,0
70 PRINT "THE FORMAT IS NOW DEFAULT"
80 GOSUB 130
90 OUT 255,8
100 PRINT "DOUBLE SIZE"
110 GOSUB 130
120 GOTO 150
130 FOR A=0 TO 2000: NEXT A
140 RET
150 REM* will revert to 64 characters/line as *
160 REM* location 16445 is ZERO"
170 END

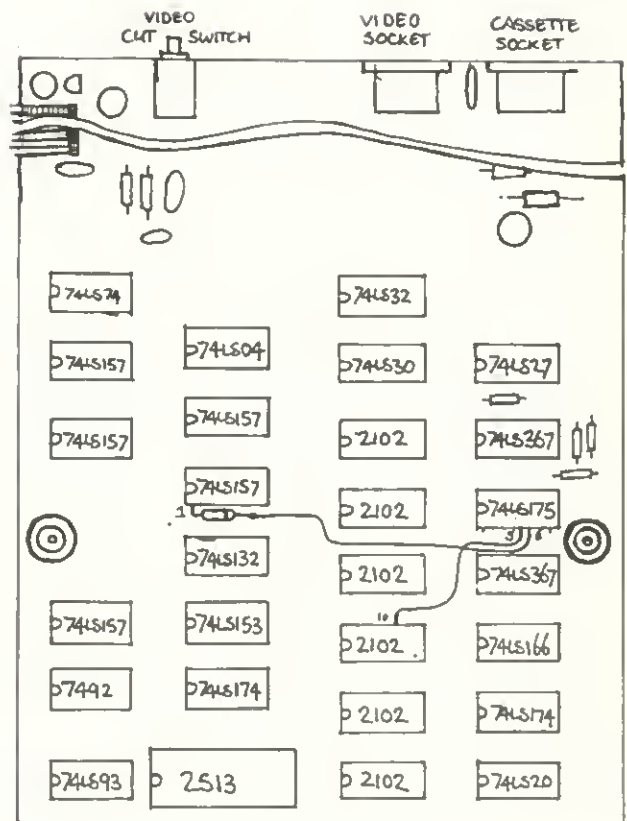
```

Continued from page 13 (Ass'y lang. prog.)

```

RLCA
RLCA
PUSH AF
LD A,E
CALL CONVRT
LD E,A
POP AF
OR E
RET
CONVRT SUB 30H
CP OAH
RET C
SUB 7
RET

```



SYSTEM 80 LAYOUT

With these 4 subroutines it should be quite easy to write assembly language programs for the TRS-80. All these types of routines and more, are available as ROM calls in the TRS-80 and SYSTEM 80 but some of these are quite involved and I recommend that the serious assembly language programmer should buy the LEVEL II ROM REFERENCE MANUAL from MICRO-80. The only two other routines needed are two I/O routines to scan the keyboard and to print information to the video display. These two can be easily accessed in the ROM, all registers are saved and the character to be printed should be in the A register, the scanning routine will return with the character in the A register, A will contain zero if no key was pressed while the keyboard was being scanned.

To scan the keyboard use : CALL 35BH

To print a character use : CALL 33AH

## LEVEL I PLUS LEVEL II - for TRS-80 Version G Board.

You will recall that this modification, as previously published, would not function on G series boards. Last month we advised those readers concerned to (a) remove the Level I chip, and (b) make good the cut p.c.b. track and (c) wait for this article. The good news is that the modified version works well, and that it looks as though it would, in fact, work on all versions of the Model I TRS-80 produced to date.

Follow all the instructions given in the June issue article, except for cutting the p.c.b. track, and observe the following changes...

\*\* The Level I ROM now plugs into Z33 and not Z34 as previously, so this will need to be inserted first, with the Level II ROM connector cable plugged into Z34, and not Z33 as previously.

\*\* Numbering anti-clockwise from the indent marker, with the pins pointing away from you, pin 20 of the Level I ROM is not to be inserted into socket Z33, but should be left "hanging out".

\*\* Switch wiring points 1 and 4 are now wired to the original intended destination of the green (enabling) Level II ROM\* wire (see illustration on page 25 of the June Issue).

\*\* Switch wiring point 5 is now wired directly to pin 20 of the Level I ROM.

It seems that some Level I units are still coming out with the "old" two-chip ROM pack. Sorry, but it starts to become impractical with these units, and if you are so blessed, we can only suggest that you either "swop" with a non-MICRO 80 reader, or bribe your local Tandy manager.

We understand that there are some Level II units coming out with a two-chip ROM pack, plugged directly into Z33 and Z34, and the same advice must apply to you, too.

\*\* REMEMBER - always power down before switching from one level to the other. There is a distinct possibility that, for a fraction of a second during change-over, both ROM sets will be enabled, and utter chaos will ensue.

P.H.

\* DECBINHEX \* by Phil Erudat.

Believing that a little education rarely goes astray, we present two gems this month dealing with the relationship between the three numeric bases used by we micro-buffs. Phil's Level I effort is well worthwhile for all you Level I'ers out there.

```

1  REM          ***** DEC BIN HEX - BY PHIL ERUDAT *****
2  REM          110 PARK ST. MOONEE PONDS VIC. (63) 370 8473
3  REM
19  U=0
20  CLS
25  P.A.17,"DECIMAL ↑ BINARY ↑ HEXIDECIMAL"
30  P."THE PROGRAM WILL CONVERT FROM DECIMAL,BINARY,OR HEX.INPUTS"
31  P."IT WILL GIVE A SINGLE PRINT OF THE INITIAL CONVERSION"
32  P."OR,A CONTINUING PRINT INCREMENTING THE DECIMAL VALUE "
33  P.
34  P=0
35  IN."1 = DECIMAL INPUT, 2 = BINARY INPUT, 3 = HEX.INPUT";P
36  P.
37  IN."DO YOU REQUIRE A CONTINUING PRINT.. 1=NO.. 2=YES ";U
39  CLS
40  ON P G.44,S00,700
43  P."SORRY THE NUMBER MUST BE BETWEEN (0 AND 65535)
44  P." "
45  IN."ENTER DECIMAL NUMBER (0 - 65535)...";0
50  IF (Q<0)+(Q>65535) CLS:G.43
55  F.X=1 TO 16:A(X)=0:N.X
57  REM * DECIMAL TO BINARY ,READ INTO AN ARRAY
60  Z=0:L=1

```

CONTINUED ON P. 24



## \*\*\*\*\* MICRO-80 PRODUCTS \*\*\*\*\*

## NEWDOS 80 IS IN STOCK.

This long-awaited disk operating system has now arrived and has already been delivered to many eager customers around Australia. It is every bit as good as it was cracked-up to be and MICRO-80 has decided to standardise on it for its own systems. Here are just a few of the many things you can do with NEWDOS 80 which is upward compatible with TRSDOS and NEWDOS + (i.e. TRSDOS and NEWDOS + programs will run on NEWDOS 80 but the reverse is not necessarily so).

- \* New BASIC commands that support variable record lengths up to 4095 bytes long.
- \* Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40, 77 or 80 track 5" mini disk drives or 8" disk drives, OR ANY COMBINATION!!
- \* A security boot-up for BASIC or machine code application programs. User never sees "DOSREADY" or "READY" and is unable to "BREAK", clear screen or issue any direct BASIC statements including "LIST".
- \* New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- \* Enhanced and improved RENUMBER that allows relocation of subroutines.
- \* Powerful program chaining.
- \* Device handling for routing to display and printer simultaneously.
- \* CDE function; simultaneous striking of the C, D and E keys will allow user to enter a mini-DOS to perform some DOS commands without disturbing the resident program.
- \* Includes Superzap 3.0 (improved, machine language version of Superzap) and all Apparat 2.1 utilities.

NEWDOS +  
 35 TRACK VERSION       \$99.00 incl. p&p  
 40 TRACK VERSION       \$110.00 incl. p&p

NEWDOS by Apparat, is the third generation disk operating system for your TRS-80. NEWDOS corrects over 70 errors and omissions in TRSDOS and Disk BASIC yet the two are completely compatible. Going from TRSDOS to NEWDOS is like going from Level I to Level II, more power, more convenience, greater speed. NEWDOS + includes the following utilities:

- \* Editor-assembler for Disk
- \* Disassembler (Z80 machine code)
- \* LM offset - allows transfers of any system tape to a Disk file - automatically relocated.

- \* BASIC 1 - Lets you convert your computer back to Level I
  - \* LVIDKSL - Saves and loads BASIC 1 programs to disk
  - \* SUPERZAP - display/print/modify any location in memory or on disk.
  - \* RENUMBER BASIC program.
- and much more.

## \*\* SCOTCH BRAND COMPUTING CASSETTES \*\*

These super-quality cassettes are now back in stock.

C-10 pack of 10	\$26.00 incl. p&p
c-30 pack of 10	\$28.00 incl. p&p

## 40 PIN EDGE CONNECTORS       \$12.30 +50c. p&amp;p

These difficult to obtain connectors are essential if you are working on a hardware project requiring connection to the TANDY bus. The connector attaches to a 40 way ribbon cable and plugs straight on to the back of your TRS-80.

## \*\*\*\*\* MICROPOLIS 77 TPACK DISK DRIVES \*\*\*\*\*

These fabulous MICROPOLIS disk drives have more than double the storage capacity of the standard 35 track drives.

## \*\* DD-7S only \$775 incl. p&amp;p \*\*

77 track MICROPOLIS drive complete with cable for four drives, power supply, chassis and includes NEWDOS 80.

## \*\* DD-7 only \$649 incl. p&amp;p \*\*

Same as above but no cable or NEWDOS 80.

## \*\* DC-4 only \$45 incl. p&amp;p \*\*

4 drive connector cable - suitable for any disk drives.

## \*\*\*\*\* DISKETTE BREAKTHROUGH \*\*\*\*\*

We have changed our brand of diskettes but not our price. We now sell genuine

## BASF DISKETTES

SINGLE SIDE/SINGLE DENSITY \$45.00 per box of 10 incl. p&p.

DOUBLE SIDE/DOUBLE DENSITY \$59.00 per box of 10 incl. p&p.

## \*\*\*\*\* SOFTWARE BY AUSTRALIAN AUTHORS \*\*\*\*\*

All our software is suitable for either the SYSTEM 80 or the TRS-80 (with the exception of Me Think it Micro Mazin vol.2). Tapes are specially recorded at high levels to ensure satisfactory loading into SYSTEM 80's.

## \*\* UTILITIES \*\*

S-KEY by Edwin Paay \$15.95 plus 50c. p&p

S-KEY is a complete keyboard driver routine for the TRS-80 and becomes part of the Level II basic interpreter. With S-KEY loaded the user will have many new features not available with the standard machine.

S-KEY features:

- \* S-KEY will provide an auto repeat for all the keys on the keyboard. If any key is held down longer than about half a second the key will repeat until it is released.
- \* Graphic symbols can be typed direct from the keyboard, this includes all 64 graphic symbols available from the TRS-80.
- \* S-KEY allows text, basic commands and/or graphics to be defined to shifted keys. This will make programming much easier as whole commands and statements can be recalled by typing shift and a letter key.
- \* Because S-KEY allows graphics to be typed directly from the keyboard, animation and fast graphics are easily implemented by typing the appropriate graphic symbols directly into PRINT statements.
- \* S-KEY will allow the user to LIST a program with PRINT statements containing graphics, properly. S-KEY does this by intercepting the LIST routine when necessary.
- \* S-KEY allows the user to list an updated list of the shift key entry's to the video display or line printer.
- \* S-KEY can be disabled and enabled when required. This allows other routines which take control of the keyboard to run with S-KEY as well.

Each cassette has TRS-80, DISK and SYSTEM 80 versions and comes with comprehensive documentation.

BMON by Edwin Paay \$19.95 plus 50 c. p&p  
THE ULTIMATE HIGH MEMORY BASIC MONITOR L2/16K and up.

Our own personnel refuse to write BASIC without first loading this amazing machine language utility program into high memory! BMON Renumbers; Displays Basic programs on the screen while they are still loading; Tells you the memory locations of the program just loaded; Lets you stop a load part-way through; Merges two programs, with automatic

renumbering of the second so as to prevent any clashes of line numbers; Recovers your program even though you did type NEW; Makes one program invisible while you work on a second (saves hours of cassette time!); Lists all the variables used in the program; Makes SYSTEM tapes; Lets you Edit memory directly...the list goes on and on. Cassette comes with 16K, 32K and 48K versions, ready to load. Can anyone afford NOT to have BMON?

## \*\* EDUCATIONAL \*\*

RPN CALCULATOR (L2/16K & L2/32K) \$24.95 +50c p&p.

Give the computer the power of a \$650 reverse polish notation calculator with 45 functions and selectable accuracy of 8 or 16 digits. The main stack and registers are continuously displayed, whilst the menu is always instantly accessible without disturbing any calculations or register values. The cassette NOW comes with both the 16K and 32K versions, the latter giving you the additional power of a programmable calculator. Comes with a very comprehensive 15 page manual, which includes instructions to load and modify the 32K programmable version to run in 16K. Whether for everyday or occasional use, this package will prove invaluable, and turn your '80 into a very powerful instrument.

TOUCHTYPE (L2/4K) \$19.95 + 50c. p&p.

An interactive, 22 lesson typing course which uses the computer's keyboard and screen to teach you to type rapidly and accurately, and a massive cassette data dump to control your progress. The computer checks for accuracy, and sets times exercises to check your progress. If you have to look at each key before you press it, or only use two fingers, then this program, plus a little perseverance, will do some amazing things to your typing speed.

## \*\* GAMES \*\*

UBOAT \$7.50 plus 50c. p&p.

Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedos, plenty of targets, working fuel and battery meters, helpful Mothership for high-sea reprovisioning, and even has emergency radio for that terrible moment when the depth-charges put your crew at risk. Requires Level II/16K.

SPACE INVADERS \$7.50 plus 50c. p&p.  
Much improved version of this arcade favourite with redesigned laser and cannon blasts, high-speed cannon, 50 roving drone targets, 10 motherships and heaps of fun for all. Level II with 4K and 16K versions on this cassette.

MMM- vol.1 and vol.2. \$7.50 each plus 50c.p&p  
Two, three-game cassettes from that master of TRS-80 graphics, Charlie Bartlett. Vol.1 brings you INDY 500, an exciting road race that gets faster and faster the longer you play, SUBHUNT in which your warship blows up unfortunate little submarines all over the place, and KNIEVEL (as in motorcycle, ramp and buses).

Vol.2 (not System 80 compatible) contains TANK, a two player battle of speed and skill, THIEF (steal an old man's hoard of loot then escape from the labyrinth before he gets you), and SHOOTOUT (a Mexican style high-noon for two).

DALEK CHASE (L2/16K and up)

\$7.50 + 50c. p&p

An absorbing game in which you save Dr Who from pursuing Daleks. The better you play, the harder it gets (and vice versa). Infinite range of play levels from HARD to UTTERLY IMPOSSIBLE, and comes complete with messages of praise and mediocre insults. (If you are really bad enough at this game, it actually refuses to play any more!) ENTHRALLING FUN FOR ALL.

\*\*\*\*\* GREEN SCREEN SIMULATOR \*\*\*\*\*

\$19.95 incl p&p

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut and curved to fit your monitor. It improves contrast and is much more restful to the eyes than the normal grey and white image. All the editorial staff of MICRO-80 (including Scrooge, Mc. Hartley) are now using GREEN SCREEN SIMULATORS on their own monitors. Please make sure to specify whether you have an old (squarish) or a new (rounded) style monitor when ordering.

\*\*\*\*\* MICRO-80 PRODUCTS \*\*\*\*\*

Use ORDER FORM on page 41

\*\*\*\*\* MPI DISK DRIVES \*\*\*\*\*

MPI is the second largest manufacturer of disk drives in the world. They use the same form of head control as on 8" drives and consequently, they have the fastest track-to-track access time available: 5 msec. All their drives are capable of single or double density operation. (Double-density operation requires the installation of a special PC board in the expansion interface. This board is not yet available in Australia). As well as the single head 40 track disk drive, MPI also make a dual-head 40 track disk drive. To all intents and purposes, a dual-head drive behaves like two single-head drives but is much cheaper.

Our MPI drives are supplied bare, set up to operate with the TRS-80. They can be used bare or a simple cabinet made up from sheet metal. They require the use of a separate power supply giving 5 volt at 0.7 amp and 12 volt at 1 amp. All drives are sold with a 90 day warranty and service is available through MICRO-80 PRODUCTS.

MPI SINGLE HEAD DISK DRIVE only \$339 incl.p&p  
MPI DOUBLE HEAD DISK DRIVE only \$449 incl.p&p  
Ring first to check availability.

\*\*\*\*\* BOOKS \*\*\*\*\*

LEVEL II ROM REFERENCE MANUAL \$24.95 plus \$1.20 p&p.

Over 70 pages packed full of useful information and sample programs. Applies to both TRS-80 and SYSTEM 80.

TRS-80 DISK AND OTHER MYSTERIES \$24.95 plus \$1.20 p&p.

The hottest selling TRS-80 book in the U.S.A. Disk file structures revealed, DOS's compared and explained, how to recover lost files, how to rebuild crashed directories - this is a MUST for the serious Disk user and is a perfect companion to any of the NEWDOS's.

LEARNING LEVEL II \$16.95 plus \$1.20 p&p.

The second hottest selling TRS-80/ SYSTEM 80 book in the U.S.A. Written by David Lien, the author of the TRS-80 Level I handbook, this book teaches you, step-by-step, how to get the most from your Level II machine.

INTRODUCTION TO T-BUG \$12.50 plus \$1.20 p&p.

This book describes in detail how to use Tandy's T-BUG monitor program. Each command is explained and discussed in detail with examples. A must for the T-BUG user.

# \*\* PROGRAMS FROM CREATIVE COMPUTING \*\*

## \*\* ADVENTURE PROGRAMS \*\*

ADVENTURELAND (L2/16K) \$14.95 + 50c. p&p  
Try to find and take treasures as you explore a fantasy world. The computer acts as your puppet and carries out your two word commands. Sometimes you will need special objects to do certain things, often a little magic is necessary. Absorbing and challenging.

THE COUNT ADVENTURE (L2/16K) \$14.95 + 50c. p&p

In this adventure, you awaken in a bed in a castle in Transylvania. You don't know why you are there but you'd better solve the puzzle before it's too late. Just as enthralling as ADVENTURELAND but blood thirstier!

## \*\* GAMES \*\*

AIR TRAFFIC CONTROLLER (L2/16K) \$7.95 + 50c. p&p

One of the hottest selling games in the USA, you are the Air Traffic Controller and the monitor is your radar screen. Bring down the aircraft safely and avoid mid-air collisions.

Z Chess (L2/16K) \$19.95 + 50c. p&p  
Seven levels of ability, contains all standard moves including castling and En Passant captures. It can play either black or white and its versatile board set-up mode allows specific positions to be played as desired.

SPACE GAMES (L2/16K) \$7.95 + 50c. p&p  
4 Space games including ULTRA-TREK, ROMULAN, STARWARS and STARLANES. Fast, real-time graphics.

STRATEGY GAMES (L2/16K) \$7.95 + 50c. p&p  
5 Strategy games including TUNNEL VISION (find your way out of a 3-D maze), EVASION - avoid the deadly snake), JIGSAW (put the puzzle together), THE MASTERS (Golf on the '80 for up to 4 players), MOTOR RACING (Compete against the computer at Indy or the Grand Prix).

GRAPHING PACKAGE (L2/16K) \$7.95 + 50c. p&p  
A set of 6 utility programs which allow you to draw BAR GRAPHS, GRAPH CARTESIAN COORDINATES, carry out POLAR GRAPHING, PARAMETRIC GRAPHING, LINEAR REGRESSION and PARABOLIC REGRESSION.

# \*\* PROGRAMS BY MICROSOFT \*\*

EDITOR ASSEMBLER PLUS (L2/16K) \$39.95 plus \$1.20 p&p  
A much improved editor-assembler and debug/monitor for L2/16K TRS-80 or SYSTEM 80. Assembles directly into memory, supports macros and conditional assembly, includes new commands-substitute, move, copy and extend.

LEVEL III BASIC \$59.95 plus \$1.20 p&p  
Loads on top of Level II BASIC and gives advanced graphics, automatic renumbering, single stroke instructions (shift-key entries) keyboard debounce, suitable for L2/16K and up (Not Disk BASIC)

ADVENTURE ON DISK \$35.95 plus \$1.20 p&p  
This is the original ADVENTURE game adapted for the TRS-80. The game fills an entire diskette. Endless variety and challenge as you seek to rise to the level of Grand Master. Until you gain skill, there are whole areas of the cave that you cannot enter. (Requires 32K One Disk)

## DISK EDITOR ASSEMBLER

SAVE \$30 ONLY \$107 plus \$2.00 p&p  
Supports Macros, linking loader, editor, cross references. See Assembly language programming article in August MICRO-80 for further details. (Requires 32K One Disk)

BASIC COMPILER \$208 plus \$2.00 p&p  
Converts Disk BASIC programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original BASIC program and is much more difficult to pirate. Note: MICROSOFT have temporarily withdrawn this program for reworking to make it less memory-hungry. The revised version should be available about the end of September. All existing owners will receive updated versions at no cost to them. Don't wait though, send in your orders now and be near the front of the queue.

## \*\*\*\*\* 16K MEMORY EXPANSION KIT \*\*\*\*\*

only \$97 incl. p&p.

These are prime, branded, 200 ns (yes, 200 nanosecond) chips. You will pay much more elsewhere for slow, 350 ns.chips. Ours are guaranteed for 12 months. A pair of DIP shunts is also required to upgrade the CPU memory in the TRS-80 - these cost an additional \$4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to instal them.



```

65 A=INT(Z/2)
70 B=Z-(A*2)
75 A(L)=B
80 Z=A
85 IF A=0 G.99
90 L=L+1
95 G.65
99 IF U>2 G.120
100 P.A.278,"(BINARY MULTIPLIERS)"
102 P.A.330,"3 1"
104 P.A.394,"2 6 8 4 2 1"
105 P.A.458,"7 3 1 0 0 0 5 2 1"
107 P.A.522,"6 8 9 9 4 2 1 5 2 6 3 1"
108 P.A.506,"8 4 2 6 8 4 2 6 8 4 2 6 B 4 2 1"
110 F.X=0 TO 63:P,"-";:N.X
115 REM * READ AND PRINT 16 BINARY DIGITS FROM ARRAY
120 P,"BINARY";
122 P.T.(8);";";
125 F.N=16 TO 1 5.-1
127 IF((INT(N/4)*4-N)=0) P,"-";";";
130 P.A(N);
135 N.N
140 P.:P.O;
145 REM *CALC.(D = 1 TO 15) FOR EACH GROUP OF 4 BINARY DIGITS
149 L=21
150 I=13:J=16
153 E=0:M=1
155 F.X=I TO J
160 T=A(X)*M
165 D=T+E
170 E=D
175 M=M*2
180 N.X
185 A(L)=D
186 L=L+1
190 IF I<1 G.200
195 G.153
197 REM* CALC.AND PRINT DECIMAL VALUE OF EACH BINARY GROUP
200 H=4096:G=0
202 F.L=21 TO 24
204 W=A(L)*H
206 P.T.(12+G);W;
208 H=H/16:G=G+13
210 N.L
212 P.
213 REM * CALC.AND PRINT HEX.DIGITS
215 P,"HEX =";
216 G=0
217 F.L=21 TO 24
220 IF A(L)>9 G.380:G.223
221 G.350:G.224
223 P.T.(15+G);A$;:G.230
224 P.T.(14+G);F;
230 G=G+13
235 N.L
240 P.
245 F.X=0 TO 64:P,"-";:N.X
246 IF U>2 G.270
250 X=16
255 F.Y=15 TO 29:5.(X,Y):N.Y
260 IF X>110 G.270
265 X=X+26
267 G.255
270 X=16
271 F.Y=30 TO 41:5.(X,Y):N.Y
272 IF X>110 G.280
273 X=X+26
274 G.271
280 IF U>2 G.320
281 X=24
285 F.Y=15 TO 32:5.(X,Y):N.Y
290 X=X+6
295 IF X=42 X=50:G.285

```

```

300 IF X=68 X=76:G.285
305 IF X=94 X=102:G.285
310 IF X=120 G.320
315 G.285
320 P.A.789,"*";:P.A.802,"*";:P.A.815,"*";
330 P.A.960,"*";
332 IF(U=2)*(Q<65535) U=U+1:Q=Q+1:G.55
334 P.A.970,"*";
335 IN."PRESS ENTER FOR NEXT CONVERSION":A$:G.20
340 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
345 REM * SUB'S TO READ HEX. VALUES
350 RESTORE
355 F.X=1 TO A(L)+1
360 READ F
365 N.X
370 RET.
380 RESTORE
385 F.X=1 TO 10 :READ K:N.X
390 F.X=11 TO A(L)+1
395 READ A$
400 N.X
405 RET.
500 REM * BINARY TO DECIMAL *
505 CLS:G.510
507 P."SORRY I CANNOT CONVERT OVER 16 DIGITS"
510 IN."HOW MANY DIGITS IN THE BINARY NUMBER (1 - 16)":H
525 IF H>16 CLS:G.507
530 CLS
535 P."EACH BINARY DIGIT IS ENTERED AS A SEPERATE INPUT"
540 F.X=1 TO 16:A(X)=0:N.X
545 B=0
550 F.X=H TO 1 S.-1
555 P.A.64,"ENTER BINARY (PRESS >ENTER< AFTER EACH DIGIT...)":
560 IN,A(X)
565 P.A.196,"BINARY":
570 P.A.204+B,A(X);
575 B=B+2
580 N.X
590 M=1:E=0
595 F.X=1 TO H
600 T=A(X)*M
605 B=B+T
610 M=M*2
615 E=B
620 P.A.204+B," =":G
625 N.X
630 Q=0
635 G.55
700 REM * HEX. TO DECIMAL *
702 CLS
705 F.X=1 TO 4:A(X)=0:N.X
710 A=10:B=11:C=12:D=13:E=14:F=15
712 G=0
715 IN."HOW MANY HEX. DIGITS ARE TO BE ENTERED":H
717 CLS
720 P."EACH HEX. DIGIT IS ENTERED AS A SEPERATE INPUT"
721 P.A.64,"PRESS >ENTER< AFTER EACH HEX. DIGIT"
723 H=5-H
725 P.A.136,"ENTER HEX. ":
730 F.X=H TO 4
733 IN,A(X)
735 P.A.210+G,"*";
740 G=G+5
745 N.X
750 M=4096:S=0
755 F.X=1 TO 4
760 L=A(X)*M
765 R=L+S
770 S=R
775 M=M/16
780 N.X
785 Q=R
790 P.A.200+G," =":R
795 G.55

```

\* JET BOAT \* by Michael Svensdotter (L1/4K)

"Just refuel your jets and away you go..."

The object of this game is to cross the river in your jet-boat without being washed over the falls. With each new crossing, the maximum allowable jet thrust will change, and as you transverse the river the current flow will vary. Work it out carefully or...over you go.

Note lines 155 and 175. If you press (ENTER) before entering a number, don't worry about the line-feed because when you do put in a number the computer will see that point (127,38) is off and will jump to line 350.

```

1 REM >> M J SVENSDOTTER
2 REM >> 4 HORTON AVE. SALISBURY NTH. 2583756
3 REM >> 2-3/80
20 CLS:P."↑↑↑↑↑↑↑↑↑↑ J E T B O A T 1111111111":P.:P.
30 Y=1:N=2
40 I."DO YOU WANT INSTRUCTIONS (Y/N) ":A:IFA=YGOS.500
80 P.AT585,:I."PRESS ENTER TO REFUEL YOUR JETS":A$
90 CLS:F.A=1T03B:P.AT79,"R E F U E L I N G J E T S "
95 F.Z=1T04:P." R E F U E L I N G J E T S ":N.Z:CLS:N.A
100 CLS:A=RND(30)+20
110 GOS.2000:P.AT23,"
120 P.AT3,"CURRENT <J>↑↑":
130 D=RND(10)+RND(0):P.AT17,A+B:
135 P=X:Q=Y
140 P.AT74,"ENTER YOUR JET ANGLE":HT954,:I.B:P.AT954
145 IFB<20B=20
150 B=D+D:IFB>85B=85
155 IFP.(127,38)=0G.350
160 P.AT70,"NOW":AT83,"THRUST":AT554,:I.C:P.AT354
165 IFC>A+3C=A+B
167 C=RND(10)+1+C:IFC<0C=0
170 P.AT70," ":AT1023,
175 IFP.(127,38)=0G.350
180 H=X:I=Y:Y=Y+A
190 GOS.1000
200 B=90-B:Y=Y-(U+C)+D
210 GOS.1000
220 X=X+(U+C)
222 G.245
225 IF(X>117)+(Y>32)+(Y<34)G.300
230 IFY>38P.AT202,"YOU'VE JUST GONE OVER THE FALLS":GOS.2200:G.80
235 IFX>117G.225
240 GOS.2200:RET.
245 K=(Y-I)/3:L=(X-H)/3:X=X-(L+3):Y=Y-(K+3)
250 F.O=1T03
255 IFO>1P=X:Q=Y
260 X=X+L:Y=Y+K:GOS.225
270 IFY<11G.340
280 N.O
290 G.130
300 P.AT282,"YOU'VE MADE IT":
320 G.330
325 P.AT282,"YOU'VE CRASHED ON THE ROCKS":
330 IFX>117X=117:GOS.2200:G.80
340 P.AT132,"YOU'RE SUPPOSED TO GO IN THE OTHER DIRECTION":G.80
350 CLS:P."MAKE SURE YOU PRESS A NUMBER BEFORE YOU PRESS ENTER
360 G.80
500 CLS:P."YOU'RE UP STREAM FROM NIAGRA FALLS":P.
510 P."A JET BOAT IS YOUR ONLY MEANS OF REACHING THE OTHER SIDE":P.
520 P."THE ONLY SAFE PLACE ON THE OTHER BANK IS NEAR THE FALLS":P.
530 P."THE MAXIMUM JET THRUST WILL BE RELAYED TO YOU":P.
540 P."THE SPEED OF THE RIVER WILL ALSO BE RELAYED":P.
550 I."PRESS ENTER TO CONTINUE":A$:CLS
560 P."MINIMUM JET ANGLE IS 20, MAXIMUM IS 90":P.
570 P."90 IS ACROSS THE CURRENT":P.
580 P."A SMALL ANGLE IS THE BEST WAY TO STAY ALIVE":P.
590 P."DUE TO ROCKS YOU MAY HIT SOME TURBULENCE":P.
600 RET.
1000 REST.
1010 F.Z=1T0B/10*2-3
1020 READU
1030 N.2
1040 RET.
1050 D..94,.91,.87,.82,.77,.71,.65,.57,.5,.42,.34,.26,.17,.09

```

```

2000 P.AT25,"UNITS OF THRUST :- MAX. 1111";A+B:AT4,
2010 F.Z=1T01000:N.Z
2020 F.X=0T0127S.127
2030 F.Y=0T030
2040 S.(X,Y)
2050 N.Y
2060 N.X
2065 P.AT040,"1111","1111","1111","1111";
2070 P.AT063,"1111";"1111" THE FALLS ";
2080 P."1111";"1111";
2090 R.(127,33):R.(127,34)
2090 Y=12:X=1
2100 O=Y:P=X
2200 F.N=1T02
2210 F.F=0T02
2220 F.G=0T02S.2
2230 F.N=1R.(P+F,0+G):R.(P+2+F,0+1):R.(P+6+F,0+1)
2235 IFY>=30G.2260
2240 IFN=2S.(X+F,Y+G):S.(X+2+F,Y+1):S.(X+6+F,Y+1)
2260 N.G
2270 N.F
2280 N.N
2300 RET.

```

\* ADDHEXAD \* by Spencer George.

This Level II program teaches hexadecimal arithmetic in Spencer's traditionally skilful method. The only processing done to this listing was to remove line-feeds from REMs to reduce space requirements. The use of separate print lines for different levels of function within this listing demonstrates that the '80 doesn't notice a linefeed in the middle of an instruction. The listing is well endowed with REMs. So no further explanations are needed here.

```

10 '      H E X A D E C I M A L   A R I T H M E T I C
20 '      COPYRIGHT      C      SPENCER GEORGE      1980
30 '      S C G      4/5/80
40 '      THIS PROGRAM IS WRITTEN IN LEVEL II
50 '      REQUIRES      9077      BYTES OF MEMORY
70 '      SPENCER GEORGE
      14 / 47 YERRIN STREET,
      BALWYN, 3103
      (03) 836 4225
100 '      THIS PROGRAM TEACHES SIMPLE HEXADECIMAL ADDITION.
      IT STARTS BY TEACHING THE SIXTEEN HEXADECIMAL
      CHARACTERS 0 1 2 3 4 5 6 7 8 9 A B C D E F
      AND THEN TEACHES SIMPLE ADDITIONS OF TWO SINGLE
110 '      HEXADECIMAL CHARACTERS.
120 '      LINES 1000 TO 1220 CONTAIN THE DATA DECKS AND MUST
      ALWAYS BE RUN.
      TESTING THE HEXADECIMAL CHARACTER SET MAY BE STARTED
      WITH GOTO1720
      TESTING THE SIMPLE ADDITIONS MAY BE STARTED WITH
130 '      GOTO 2000      (AFTER RUN      THEN      BREAK)
1000 '      PRINTS A LARGE HEADING
      H E X A D E C I M A L   A R I T H M E T I C
1010 CLS:
      PRINT$S12, CHR$(23): "HEXADECIMAL ARITHMETIC"
1020 FOR J = 1 TO 2003:
      NEXT J
1030 '      DATA DECK FOR ASSORTED NUMBERS A TO F
1040 DIM Y(12)
1050 FOR J = 1 TO 12
1060 READ Y(J)
1070 NEXT J
1080 DATA 10,14,15,13,12,11,15,13,11,12,14,10
1090 '      VIDEO POSITIONS FOR HEXADECIMAL CLOCK
1100 DATA 106,178,240,300,310,637,762,819,874,800,731,599,469,343,219,161
1110 DIM N2(15), N3(15)
1120 FOR I = 0 TO 15
1130 READ N2(J)
1140 NEXT J
1150 '      HEXADECIMAL COMBINATIONS WITH A TOTAL OF 10

```



```

1160 FOR J = 0 TO 5
1170 READ N3$(J)
1180 NEXT J
1190 DATA A + 5 = 10, C + 4 = 10, E + 2 = 10, B + 5 = 10, D + 3 = 10, F + 1 = 1
0
1200 '      HEXADECIMAL NUMBERS  0  TO  1F
1210 DIM H$(31):
      FOR J = 0 TO 31:
        READ H$(J) :
        NEXT J:
        DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,11,12,13,14,15,16,17,1D,19,1A,1D,1C,1
D,1E,1F
1220 '      THE 16 HEXADECIMAL CHARACTERS DISPLAYED
1230 CLS
1240 PRINT@15,"HEXADECIMAL CHARACTERS ARE : "
1250 FOR J = 0 TO 15
1260 PRINT@96D-J*64,H$(J);
1270 FOR K = 1 TO 200 :
      NEXT K

1280 NEXT J
1290 FOR K = 1 TO 1200 : NEXT K
1300 '      16 HEXADECIMAL CHARACTERS DISPLAYED
      AND STUDENT IS ASKED TO TYPE THESE CHARACTERS
      BUT THE ANSWER IS NOT CHECKED
1310 CLS
1320 PRINT@64,"THE FIFTEEN HEXADECIMAL CHARACTERS ARE
0 1 2 3 4 5 6 7 8 9 A B C D E F"
1330 PRINT@440,"PLEASE TYPE IN ORDER THE FIFTEEN HEXADECIMAL CHARACTERS
TYPE  C N T E R  AT THE END OF THE LIST
AND A BLANK SPACE BETWEEN CHARACTERS

.
1340 INPUT AAS
1350 '      THE 8 ODD HEXADECIMAL CHARACTERS ARE DISPLAYED
1360 CLS
1370 PRINT"ODD HEXADECIMAL CHARACTERS ARE:
.
1380 FOR J = 1 TO 15 STEP 2
1390 PRINT TAB(J*2); H$(J)
1400 FOR K = 1 TO 200:
      NEXT K
1410 NEXT J
1420 FOR K = 1 TO 120 :
      NEXT K
1430 '      THE ODD HEXADECIMAL CHARACTERS ARE DISPLAYED
      AND THE STUDENT IS ASKED TO TYPE THE LIST
      BUT THE LIST IS NOT CHECKED
1440 CLS
1450 PRINT@64,"THE ODD HEXADECIMAL CHARACTERS ARE

1 3 5 7 9 B D F

.
1460 PRINT"PLEASE TYPE THE ODD HEXADECIMAL CHARACTERS
TYPE  C N T E R  AT THE END OF THE LIST
AND A BLANK SPACE AFTER EACH CHARACTER

1470 INPUT AAS
1480 '      THE 8 EVEN HEXADECIMAL CHARACTERS ARE DISPLAYED
1490 CLS
1500 PRINT"EVEN HEXADECIMAL CHARACTERS ARE :
.
1510 FOR J = 0 TO 14 STEP 2
1520 PRINTTAB(J*3); H$(J)
1530 FOR K = 1 TO 200 :
      NEXT K
1540 NEXT J
1550 FOR K = 1 TO 1200 :
      NEXT K

```

```

1550      THE EVEN HEXADECIMAL CHARACTERS ARE DISPLAYED
      AND THE STUDENT IS ASKED TO TYPE THE LIST
      BUT THE LIST IS NOT CHECKED
1570 CLS :PRINT@64,"THE EVEN HEXADECIMAL CHARACTERS ARE
0 2 4 6 8 A C E
1580 PRINT" PLEASE TYPE THE EVEN HEXADECIMAL CHARACTERS

DEPRESS     E N T E R     AT THE END OF THE LIST
AND PLACE A BLANK SPACE BETWEEN CHARACTERS

1590 INPUT A$
1600 '      THE DECIMAL EQUIVALENTS OF THE 16 HEXADECIMAL CHARACTERS      ARE
      DISPLAYED
1610 CLS
1620 PRINT"DECIMAL EQUIVALENTS FOR HEXADECIMAL CHARACTERS ARE :
'
1630 GOTO 1710
1640 '      A SUBROUTINE WHICH DISPLAYS THE 16 HEXADECIMAL
      CHARACTERS AND THEIR DECIMAL EQUIVALENTS
1650 FOR JA = 15 TO 0 STEP -1
1660 PRINT
1670 PRINTH$(JA); "      H = "; JA,
1680 FOR K = 1 TO 200 :
      NEXT K
1690 NEXT JA
1700 RETURN
1710 GOSUB 1650
1730 '      THE STUDENT IS TESTED ON THE DECIMAL EQUIVALENTS
      FOR THE HEXADECIMAL CHARACTERS A TO F
1734 FOR J = 1 TO 12
1735 IF RE = 9 THEN CLS:
      GOTO 1750
1740 IF J < 6 AND J <> 1 THEN GOSUB 1650 ELSE IF J > 1 CLS
1750 PRINT@94,"GIVE THE DECIMAL EQUIVALENT FOR":
1760 PRINT@150,"THE FOLLOWING HEXADECIMAL NUMBER":
1770 PRINT@206, H$( Y(J) ); " H = ";
1780 N = Y(J)
1790 INPUT N2
1800 '      IF RESPONSE IS INCORRECT THEN THE CORRECT
      RESPONSE IS DISPLAYED SLOWLY 10 TIMES
      AND A PRINTED RECORD OF CORRECT RESPONSE
      AND GIVEN ANSWER IS PRODUCED
1810 IF N2 <> Y(J) THEN CLS :
      PRINT CHR$(23);:
      FOR K = 1 TO 12:
      PRINT H$(K); " H = "; K:
1820 IF N2 <> Y(J) THEN FOR L = 1 TO 500:
      NEXT L:
      NEXT K:
      PRINT CHR$(20):
      PRINT@960, H$(N); " H = ";
1830 IF N2 <> Y(J) AND PEEK(14312) = 63 THEN
      LPRINT CHR$(14); "THE DECIMAL EQUIVALENT OF "; H$(N); " IS "
; N ; CHR$(15); " NOT ";N2
1831 IF N2 <> Y(J) THEN 1790
1840 '      WHEN THE ANSWER IS CORRECT THE STUDENT IS REWARDED
      BY LARGE WELL DONE APPEARING ON THE SCREEN
1850 CLS:
      FOR U = 1 TO 37:
      PRINTCHR$(23); " WELL DONE ";:
      FOR UU = 1 TO 30:
      NEXT UU :
      NEXT U :
      FOR U = 1 TO 200 :
      NEXT U :
      CLS
1860 NEXT J
1870 '      A LARGE CLOCKFACE WITH THE 16 HEXADECIMAL
      CHARACTERS IS DISPLAYED

```

```

1800 CLS:
      PRINT@64,"HEXADECIMAL CHARACTERS";:
      GOSUB 1950
1890 FOR J = 1 TO 2400 :
      NEXT J
1900 CLS
1910 '      THE IMPORTANT TOTALS  A + 5 = 10
                                B + 5 = 10
                                C + 4 = 10
                                D + 3 = 10
1915 '      E + 2 = 10
                                F + 1 = 10

1920 PRINTCHR$(23):
      PRINT"REMEMBER
':
FOR J = 0 TO 5:
      PRINT N3$(J) :
      FOR K = 1 TO 800 :
      NEXT K:
      PRINT:
      NEXT J :
      FOR J = 1 TO 1200 :
      NEXT J
1930 GOTO 2010
1940 '      A SUBROUTINE FOR PRODUCING THE HEXADECIMAL CLOCKFACE
1950 CLS
1960 FOR JJ = 0 TO 15
1970 PRINT@N2(JJ),H$(JJ);
1980 NEXT JJ
1990 RETURN
2000 '      POSITIONS FOR THE COUNTING ARROWS ON THE
      HEXADECIMAL CLOCKFACE
2010      N3(0) = N2(0) + 64
2020 N3(8) = N2(8) - 64
2030 FOR J = 1 TO 7 :
      N3(J) = N2(J) - 4 :
      NEXT J
2040 FOR J = 9 TO 15 :
      N3(J) = N2(J) + 4 :
      NEXT J
2050 '      PRODUCES 10 HEXADECIMAL ADDITIONS FOR SINGLE CHARACTER
      NUMBERS.
      THE TOTAL WILL ALWAYS BE GREATER THAN 9.
      FOR THE FIRST 5 QUESTIONS THE CLOCKFACE WILL BE VISIBLE.
2060 FOR JR = 1 TO 10
2070 R1 = RND(14) + 1
2080 R2 = RND(14) + 1
2090 IF R1 + R2 < 10 THEN 2070
2100 CLS
2110 IF JR < 6 GOSUB 1950
2120 PRINT@0,"WHAT IS THE HEXADECIMAL";
2130 PRINT@64,"SUM OF";
2140 PRINT@128,H$(R1); " + "; H$(R2); " = ";
2150 INPUT 2$
2160 '      WHEN THE ANSWER IS CORRECT THE STUDENT IS
      REWARDED BY CORRECT APPEARING 10 TIMES ON
      THE SCREEN
2170 IF 2$ = H$(R1+R2) THEN CLS:
      FOR U = 1 TO 12:
      PRINT CHR$(23); "CORRECT" :
      NEXT U :
      FOR U = 1 TO 200 :
      NEXT U :
      CLS :
      RR = RR + 1:
      GOTO 2380
2180 '      WHEN AN INCORRECT RESPONSE IS MADE THE CORRECT
      SOLUTION IS DISPLAYED AND PRINTED
2190 PRINT
2200 PRINTH$(R1); " + "; H$(R2);
2210 IF PEEK(14312) = 63 THEN LPRINT H$(R1); " + "; H$(R2);

```

```

2220 IF R2 > R1 THEN S = R2 :
      R2 = R1:
      R1 = S:
      PRINT " = "; H$(R1); " + "; H$(R2) :
      IF PEEK(14312) = 63 THEN
        LPRINT " = "; H$(R1); " + "; H$(R2):
2230 IF R1 + R2 > 16 THEN PRINT TAB(5); " = "; H$(R1); " + "; H$(16 - R1);
      " + "; H$(R2 + R1 - 16) :
      PRINT TAB(5); " = 10 + "; H$(R2 + R1 - 16)
2240 IF R1 + R2 > 16 AND PEEK(14312) = 63
      THEN LPRINT TAB(5); " = "; H$(R1); " + ";
        H$(16 - R1); " + "; H$(R2 + R1 - 16):
        LPRINT TAB(5); " = 10 + "; H$(R2 + R1 - 16)
2250 PRINT TAB(5); " = "; H$(R1 + R2)
2260 IF PEEK(14312) = 63 THEN
      LPRINT TAB(5); " = "; H$(R1 + R2) ; "
      " ; Z$
2270 GOSUB 1950
2280 " WHEN THE ANSWER IS INCORRECT A CLOCKFACE IS SHOWN
      AND ARROWS MOVE AROUND IT (IN A CLOCKWISE FASHION)
      TO SHOW THE CORRECT SUM
2290 FOR J = 0 TO R2 :
      FOR TE = 1 TO 100:
      NEXT TE:
      X = R1 + J :
      IF X > 15 THEN X = X - 10
2300 IF X > 0 AND X < 8 PRINT@B(X)-3, " " :CHR$(91); :
      GOTO 2350
2310 IF X > 8 PRINT@B(X) (3,CHR$(93)) :
      GOTO 2350
2320 IF X = 0 PRINT@B(X), CHR$(91): ELSE PRINT@B(X), CHR$(92):
2330 NEXT J
2360 PRINT@360, " ";
2370 GOSUB 2440
2380 NEXT JR
2390 RE = 9:
      CLS:
      PRINT@128, "WOULD YOU LIKE ANOTHER SET OF QUESTIONS ON HEXADECIMAL
      CHARACTERPS ?
      THEN TYPE C ENTER
      WOULD YOU LIKE ANOTHER SET OF QUESTIONS ON HEXADECIMAL ADDITION THEN TYPE C
      ENTER
2391 INPUT Z$:
      IF Z$ = "C" THEN 1720
2392 IF Z$ = "A" THEN 2000 ELSE CLS
2400 PRINT@520,CHR$(23); "GOOD BYE"
2410 END
2430 " A SUBROUTINE TO FLASH
      PLEASE TOUCH SPACE BAR WHEN READY
2440 FOR JH = 1 TO 25
2450 X$ = INKEY$
2460 IF X$ <> " " RETURN
2470 PRINT@970, "PLEASE TOUCH SPACE BAR WHEN READY";
2480 NEXT JH
2490 FOR JH = 1 TO 25
2500 X$ = INKEY$
2510 IF X$ <> " " RETURN
2520 PRINT@970, " ";
2530 NEXT JH
2540 GOTO 2440
4000 LPRINT CHR$(14)
4010 FOR J = 1 TO 8
4020 LPRINT " SIMPLE HEXADECIMAL ADDITION"
4030 NEXT J
4040 LPRINT "

```



```

4050 FOR J = 1 TO B
4060 LPRINT'S C G           4 MAY, 1980
4070 NEXT J
4080 LPRINT CHR$(15)
4090 LPRINT'

```

COPYRIGHT C SPENCER GEORGE 1980

# B GAME (L2/16K) - by Lance Lawes.

Now here's an idea! A game for businessmen and executives that they CAN play at work - claiming that they need some extra practice!

If you have any ideas about going into the gizmo business, then this program will soon discover if you have what it takes! Each section of program logic is named, so no notes will be required - perhaps you might like to play with some of the random generators though, just to make the opposition a little tougher?

```

5 ' ***** BGAME (L2/16K) - BY LANCE LAWES *****

10 RANDOM:CLS:CLEAR100:DEFSTRF:DEFDBLA,C,L,R:DEFINTP,Q,X,M,O
20 F1="#####":A1=10000:A2=500000:P=70:Q=1500:A3=105000:A4=100000
30 L1=350000:L2=165000:L3=200000:I=.0125:FC'=120000
40 PRINT@276,"BUSINESS MAN"
41 PRINT@340,STRING$(24," ")
42 PRINT@440,":::INPUT'DO YOU WANT INSTRUCTIONS (Y/N) ";X$
43 IFX$<>"Y"ANDX$<>"N"GOTO42
44 IFX$="Y"GOTO2050
100 FOR N=1 TO 12
105 GOSUB1000 'PRINTOUT BALANCE SHEET
110 INPUT'PRESS <ENTER> WHEN READY ";X
140 CLS
150 A1=A1+A2-L1:A2=0:L1=0 'SETTLEMENT OF DEBTORS & CREDITORS
160 CC=A55(L2+J):C3=C3+CC:L2=L2+CC 'INTEREST CALCULATION
170 LB=0:INPUT'LOAN REPAYMENTS.....";LD:L2=L2-LB:A1=A1-LB

180 LB=D:INPUT'LOAN FUNDS OBTAINED.....";LB:L0=LB+L2:IF L0>SD
0000 GOTO181 ELSE185
181 PRINT'YOU HAVE EXCEEDED YOUR CREDIT LIMIT":L0=0:GOTO180
185 L2=L2+LB:A1=A1+LB
190 AD=0:INPUT'MACHINERY PURCHASED.....";AD:A4=A4+AD:A1=A1-AD

200 AD=D:INPUT'MACHINERY SOLD.....";AD:A4=A4-AD:A1=A1+AD

210 FC'=120000+100000/A4:C2=C2+FC':A1=A1-FC'
220 ' CALCULATE BUYERS MARKET
230 MO=AND(15000):IF MO<8000 GOTO230
240 OP=AND(100):IFOP<75 GOTO240
250 INPUT'HOW MANY UNITS DO YOU WISH TO PURCHASE.....";Q1:IFQ1>5000 PRINT'YOU HA
VE EXCEEDED YOUR WAREHOUSE STORAGE CAPACITY":GOTO250
260 INPUT'HOW MUCH PER UNIT ARE YOU PREPARED TO PAY..";P1
270 O0=AND(12000):IF O0>MO GOTO270
280 IFP1<OP THEN 300 ELSE 350
300 ' WHEN THE OPPOSITION BEATS US ON PRICE
310 Q2=MO-O0:IFQ2>Q1 THEN Q2=Q1
315 O=O+Q2
320 AC=P1*Q2:A3=A3+AC:L1=L1+AC
330 PRINT'YOU HAVE PURCHASED ";Q2;" UNITS @ ";P1;" EACH FOR ";AC
335 PRINT'YOU HAVE ";O;" UNITS OF INVENTORY ON HAND"
340 GOTO380
350 ' WHEN WE DEAT OR MATCH THE OPPOSITION ON PRICE
360 Q2=Q1:GOTO315
380 ' CALCULATE SALES MARKET
390 MO=AND(15000):IF MO<8000 GOTO 390

```

```

400 OP=RND(100):IFOP<50 GOTO400
410 OQ=RND(12000):IFOO>MO GOTO410
420 Q1=0:INPUT"HOW MANY UNITS HAVE YOU TO SELL.....":Q1
430 P1=0:INPUT"WHAT IS YOUR BEST PRICE.....":P1
440 IFP1>OP THEN 450 ELSE 500
450 " WHEN THE OPPOSITION BEATS US ON PRICE
460 Q2=MO-OQ:IFQ2>Q THEN 461
461 IFQ2>Q1 THEN Q2=Q1
462 IFQ2>Q THEN Q2=0
470 RA=Q2*P1:R1=R1+RA:A2=A2+RA
480 GOTO525
500 " WHEN WE BEAT THE OPPOSITIONS PRICE
510 Q2=Q1:IF Q2>Q THEN Q2=Q
520 GOTO470
525 P=0:P=A3/O:CA=O2*P:C1=C1+CA:A3=A3-CA:G=O-O2
530 PRINT"YOU HAVE SOLD ":Q2:" UNITS @ ":P1:" EACH FOR ":RA
540 PRINT"AT A COST OF ":P;" EACH FOR ":CA
545 PRINT"YOU HAVE ":Q;" UNITS OF INVENTORY ON HAND"
550 C4=R1-C1-C2-C3:L4=L4+RA CA-CC-FC
551 L0=0:INPUT"DIVIDEND PAID..... ":L0:L4=L4-L0:A1=A1-L
0
570 INPUT"PRESS <ENTER> WHEN READY..... ":X
580 IFA1<0GOTO2000 ELSE 585
585 NEXTN
590 GOTO2025
1000 CLS "PRINTOUT OF BALANCE SHEET
1010 PRINT"CASH FUNDS",:PRINTUSINGF1:A1,:PRINT" CREDITORS ",:PRINTUSINGF1:L
1
1020 PRINT"DEBTORS",:PRINTUSINGF1:A2,:PRINT" LOAN FUNDS",:PRINTUSINGF1:L2
1030 PRINT"INVENTORY",:PRINTUSINGF1:A3,:PRINT" CAPITAL",:PRINTUSINGF1:L3
1040 PRINT"MACHINERY",:PRINTUSINGF1:A4,:PRINT" PROFITS",:PRINTUSINGF1:L4
1050 A5=A1+A2+A3+A4:L5=L1+L2+L3+L4
1060 PRINT"ASSETS",:PRINTUSINGF1:A5,:PRINT" LIABILITIES",:PRINTUSINGF1:L5
1070 PRINTSTRING$(E3,"-")
1080 PRINT"VARIABLE COST",:PRINTUSINGF1:C1,:PRINT" SALES",:PRINTUSINGF1:R1
1090 PRINT"FIXED COSTS",:PRINTUSINGF1:C2
1100 PRINT"INTEREST",:PRINTUSINGF1:C3
1110 PRINT"PROFITS",:PRINTUSINGF1:C4
1120 PRINTTAB(16)STRING$(16,"-");PRINTTAB(40)STRING$(15,"-")
1130 PRINT",:PRINTUSINGF1:R1,:PRINT" ",:PRINTUSINGF1:R1
1140 PRINTSTRING$(E3,"-")
1150 RETURN
2000 GOSUB1000
2010 PRINT"YOU HAVE BECOME INSOLVENT AND YOUR BUSINESS OPERATION
HAS BEEN CLOSED DOWN PENDING THE APPOINTMENT OF A LIQUIDATOR"
2020 GOTO2070
2025 GOSUB1000
2030 PRINT"YOU HAVE COMPLETED A YEAR'S TRADING. YOU MAY NOW HAVE YOUR
BOOKS AUDITED AND YOUR TAXATION RETURN PREPARED"
2040 GOTO2040
2050 CLS
2055 PRINT:PRINT:PRINT:PRINT
2060 PRINT"THIS IS A PROGRAM TO TEST YOUR SKILL AS A BUSINESSMAN
INITIALLY. YOU ARE SUPPLIED WITH A PREDETERMINED BALANCE
SHEET. YOU ARE BUYING AND SELLING A SINGLE PRODUCT BUT YOU ARE
OPERATING IN A COMPETITIVE MARKET, AND YOUR COMPETITOR WILL"
2070 PRINT"STOP AT NOTHING TO UNDERCUT YOU ON PRICE.

YOU WILL BE GIVEN TWELVE (12) MONTHS IN WHICH TO MAKE A PROFIT
THE RULES OF TRADING ARE AS FOLLOWS:-"
2080 PRINT:INPUT"PRESS <ENTER> TO CONTINUE ":X:CLS
2090 PRINT:PRINT"CASH MANAGEMENT
1) ALL PURCHASES & SALES WILL BE ON A CREDIT BASIS. ALL DEBTORS
ACCOUNTS WILL BE RECEIVED. AND ALL CREDITORS ACCOUNTS WILL BE
PAID, IN THE FOLLOWING MONTH.
2) FIXED EXPENSES, AND ANY MACHINERY PURCHASED, WILL BE
2100 PRINT"PAID FOR IN THE CURRENT MONTH
3) YOU MUST REMAIN SOLVENT AT ALL TIMES (ABSOLUTELY NO BANK
OVERDRAFT). HOWEVER, IF YOU RUN SHORT OF FUNDS, YOU MAY BORROW
BUT INTEREST IS PAYABLE AT 15% P.A. ON THE BALANCE OUTSTANDING"
2110 PRINT"AT THE BEGINNING OF THE MONTH. SHOULD YOU BE INSOLVENT ATTHE END OF A
MONTH YOUR BUSINESS OPERATION WILL BE IMMEDIATELY
CLOSED DOWN PENDING THE APPOINTMENT OF A LIQUIDATOR"
2120 PRINT:INPUT"PRESS <ENTER> TO CONTINUE ":X:CLS

```

```

2130 PRINT:PRINT"MARKET FORCES
YOU ARE COMPETING IN THE MARKET PLACE FOR THE PURCHASES & RESALE OF A SUPA DUPA G
ISMO THAT NO MORON SHOULD BE WITHOUT
THE LAW OF SUPPLY & DEMAND APPLIES, SO YOU WILL GET MARKET
PREFERENCE ONLY WHEN"
2140 PRINT"A) YOU OFFER YOUR SUPPLIERS A HIGHER PRICE FOR YOUR
MERCHANDISE THAN YOUR OPPOSITION
B) YOU CAN UNDERCUT YOUR OPPOSITION'S SELLING PRICE"
2150 PRINT:INPUT"PRESS <ENTER> TO CONTINUE ";X:CLS
2160 PRINT:PRINT:PRINT"YOUR INITIAL INVENTORY IS 1500 UNITS VALUED AT COST
PRICE AT $70 EA.
AVERAGE UNIT SELLING PRICE IS ABOUT $100 EA.
AVERAGE SALES ARE ABOUT $500,000 PER MONTH"
2170 PRINT"VARIABLE COSTS ARE ABOUT 70% OF SALES, BUT THIS MAY
VARY DEPENDING ON HOW WELL YOU HAGGLE PRICES WITH SUPPLIERS
FIXED COSTS ARE PRESENTLY $120,000, BUT THIS MAY BE REDUCED
BY UPGRADING YOUR OUTDATED MACHINERY WITH MORE MODERN"
2180 PRINT"EQUIPMENT WHICH IS URGENTLY NEEDED"
2190 PRINT:INPUT"PRESS <ENTER> WHEN READY TO START TRADING ";X:CLS:GOTO100

```

# ► GREGORIAN CALENDAR \*

very level. If user will want this little gem in their collection - the program is fully self-  
prompting and generates a full calendar for any year that's ever likely to be useful to you.

```

10 REM. THE GREGORIAN CALENDAR
... B.J.C. 1979. (C) MICRO-80 1980.
20 DEFINT N,Y,D,J,I,M,K,W: POKE16553,255: XX=0
30 DIM A$(12),N(12),W(7)
40 GOSUB 90
50 GOSUB 610
60 D=H6
70 GOSUB 250
80 END
90 FOR J=1 TO 12
100 READ A$(J),N(J)
110 DATA"J A N",31,"F E B",28,"M A P",31,"A P R",30
120 DATA"M A Y",31,"J U N",30,"J U L",31,"A U G",31
130 DATA"S E P",30,"O C T",31,"N O V",30,"D E C",31
140 NEXT J
150 IF XX=> 1 THEN 180
160 CLS: PRINT CHR$(23): INPUT "
ENTER YEAR NUMBER - ";Y
170 REM. F = NORMAL LEAP YEAR ROUTINE
FF= CENTURY LEAP YEAR ROUTINE
180 FF=Y/400
190 IF INT(FF)-FF=0 THEN 220
200 FF=Y/100
210 IF INT(FF) FF=0 THEN 240
220 F=Y/4
230 IF INT(F)-F=0 THEN N(2)=N(2)+1
240 CLS: PRINT @ 450,CHR$(23) "CALENDAR FOR ";Y
250 RETURN
260 FOR M=1 TO 12
270 FOR X=1 TO 2000: NEXT: CLS
280 PRINT @ 77,Y: PRINT @ 106,Y: PRINT: PRINT TAB(24)* -- ";A$(M);" --"
290 PRINT TAB(14) "SUN MON TUE WED THU FRI SAT": REM.
2 SPACES BETWEEN DAY NAMES
300 D1=D
310 FOR I=1 TO N(M)
320 W(D)=I
330 D=D+1
340 IF D <= 7 THEN 460
350 K=1
360 IF K=D1 THEN 400
370 PRINT TAB(14)* ";; REM. 5 SPACES BETWEEN "
380 K=K+1
390 GOTO 360

```

```

400     IF D1 > 6 THEN 450
410     IF W(D1) < 10 PRINT TAB(14)" ";:
        REM. 1 SPACE BETWEEN " "
420     PRINT TAB(14) W(D1);" ";: REM. 1 SPACE BETWEEN " "
430     D1=D1+1
440     GOTO 400
450     IF W(D1) < 10 PRINT TAB(14)" ";W(7):GOTO 470:
        REM. 1 SPACE BETWEEN " "
460     PRINT TAB(14) W(7)
470     D=1: D1=1
480     NEXT I
490     IF D=1 THEN 560
500     K=1
510     IF K=D-1 THEN 550
520     PRINT TAB(14) W(K);" ";: REM. 1 SPACE BETWEEN " "
530     K=K+1
540     GOTO 510
550     PRINT TAB(14) W(D-1)
560 NEXT M
570 FOR X=1 TO 2500: NEXT
580 RESTORE: XX=XX+1: Y=Y+1: GOTO 40
590 RETURN
600 REM.      ROUTINE TO CALCULATE DAY-NAME OF 1ST JAN.
610 M=1: D=1: R=INT(.6+(1/M)): S=Y-R: T=1+12*R
620 U=S/100: H=INT(U/4): H1=INT(U): H2=INT((S*S)/4)
630 H3=INT(13*(T+1)/5): H5=H3+H2-H1+H
640 H6=H5-(7*INT(H5/7))+1
650 RETURN

```

# ANDROID SHOOTMAN (L2/16K)

This game is really a variation on hangman, but instead of the gallows being built up bit by bit, your executioner is, until you are finally shot!

Subroutine 1000 reads the words in the data statements into the N\$ array. There can be up to 500 words, as long as the last word is 'END'.

Subroutine 1050 pokes you, the right hand man, onto the screen. It uses the data in line 1500 for this.

One of the words read into the array is chosen, and this is stored in W\$. A\$ contains either dashes (chr\$(95)) or letters that have been guessed. G\$ contains all the letters you have guessed so far. Line 45 checks to see if you have already guessed the letter, while lines 50-70 check to see if the letter is in the word you are guessing. If it is, it replaces the dash in A\$ with the correct letter (line 60, A\$ = LEFT\$ ----)

If a guess is incorrect, the program jumps to line 100, where the wrong guesses (WG) is incremented. The executioner and gun is stored at 110 to 160. If you feel 5 chances is not enough, this may be increased by breaking the POKES into more lines, ending each with a 'GOTO 20' and adding the appropriate line NO's in the ON WG GOTO -- statement.

When all your chances are up, the program goes to line 170. 170 turns the R.H.S. man's head. 180 turns the executioner's head, 210 warms up the gun, and 220 shoots you. Line 235 blasts you to bits by poking you on the screen in reverse order. The program then prints the correct word, and waits for you to hit ENTER to continue. This happens after you guess the word correctly too.

```

0 ' ***** ANDROID SHOOTMAN (L2/16K) - BY D. J. MOORHOUSE *****
1 '      315 EUREKA STREET , BALLARAT , VIC. 3350
2 '      GRAPHICS BY THE AUTHOR OF 'ANDROID HIM'

```

```

3 CLEAR3000: DIM A$(100), N$(500)
5 CLS: GOSUB1050: GOSUB1000: G$="": CN=0: WG=0
6 RESTORE: GOSUB1050: G$="": CN=0: WG=0
10 RANDOM: B=RND(A): W$=N$(B): L=LEN(W$): A$=STRING$(L,95): CN=0
20 PRINT@65,"YOUR WORD IS ";A$
30 PRINT@129,"INPUT YOUR GUESS ";: INPUT B$: B$=RIGHT$(B$,1)
44 F=0
45 FOR G=1 TO LEN(G$): IF B$=MID$(G$,G,1) THEN PRINT@128,"ALREADY USED THAT!": G=200: F=-
1

```



```

46 NEXTG: IFFTHENFORG=AT0500:NEXT:GOTO20
47 G$=G$+B$
48 PRINT@I93,"LETTERS USED ";G$
50 C=0:FORX=1TOL:IFB$(X)MID$(W$,X,1)THEN70
60 C=1:CN=CN+1:A$=LEFT$(A$,X-1)+B$+RIGHT$(A$,L-X)
70 NEXTX:IFC=0THEN100
80 IFCN=LTHENPRINT@6S,"YOUR WORD IS ";W$:PRINT@257,"CONGRATULATIONS!! YOU GOT TH
E WORD WITHOUT A KILLING"ELSE20
90 INPUTE$:CLS:GOTO6
100 WG=WG+1:ONWGGO110,120,130,150,160,170
110 POKE16201,160:POKE16202,176:POKE16203,149:POKE16204,170:POKE16205,176:POKE16
206,144:GOTO20
120 POKE16075,191:POKE16076,175:POKE16139,159:POKE16140,175:GOTO20
130 POKE16073,160:POKE16074,156:POKE16137,139:POKE16138,160
140 POKE16077,172:POKE16078,144:POKE16141,184:POKE16142,135:GOTO20
150 POKE16010,190:POKE16011,155:POKE16012,167:POKE16013,189:POKE15947,148:POKE15
948,160:GOTO20
160 POKE16141,128:POKE16142,120:POKE16077,140:POKE16070,140:POKE16079,140:POKE16
080,140:POKE16001,62:POKE16002,45:POKE16003,61:GOTO20
170 POKE15982,169:POKE15983,128:POKE16045,184:POKE16046,137:POKE16047,191:POKE16
048,191:POKE16049,132
180 FORG=0T01000:NEXTG
190 POKE16009,136:POKE16010,191:POKE16011,191:POKE16012,183:POKE15013,100:POKE15
940,140:POKE15947,120
200 FORG=0T01000:NEXT
210 FORG=1T030:FORH=1T010:NEXTH:POKE16003,128:FORH=1T010:NEXTH:POKE16003,61:NEXT
220 FORG=16003T016090:POKEG,61:NEXT:FORG=16091T016100:POKEG,61:POKEG=8,128:NEXT:
FORG=16100T016109:POKEG,120:NEXT
230 RESTORE
235 FORT=5T01STEP-1:READD,C,D,E,F,G:K=T*64+15916:POKEK,B:POKEK+11,C:POKEK+2,D:PO
KEK+3,E:POKEK+4,F:POKEK+5,G:NEXT
240 PRINT@256,"THE WORD WAS ";W$
250 GOTO90
1000 A=1
1010 READN$(A):IF N$(A)="END" THENRETURN
1020 A=A+1:GOTO1010
1050 FORT=1T05:K=(T*64)+15916
1060 READD,C,D,E,F,G
1070 POKEK,B:POKEK+1,C:POKEK+2,D:POKEK+3,E:POKEK+4,F:POKEK+5,G
1080 NEXTT:RETURN
1500 DATA120,120,146,160,120,120,126,190,155,167,189,120,160,156,191,191,172,144
,139,190,159,175,104,135,160,176,149,170,176,144
2000 DATAGUNPOWDER,HORNSNOGGLE,UTILITY,MANUSCRIPT,ROCKET,URANIUM,ELEPHANT,OXYGEN
,COMPUTER,INDIAN,MATHEMATICS,SPY,BOMBING,CONTROVERSIAL,ALLIGATOR,CROCODILE
2002 DATAAUSTRALIA,THRUST,KEROSINE,CAFRICIOUS,GAUCHE,SPELLING,WINTER,ANDROID,ROC
KET,SPAGHETTI,ROSTER,MAGNESIUM,ALPHA,TRINOMIAL,WINDOW,HYPERGEOMETRIC,POISSON,QUA
DRATIC,CARPET,CASSETTE
2004 DATACEILING,WAX,RHYTHM,AUTUMN,LIBRARY,MNEMONIC,PNEUMONIA,PSYCHIATRIST,HAEMO
RHAGE,TOUCH
2005 DATAESTABLISHMENT,SCIENTIFIC,EXTRAORDINARY,OSCILLOSCOPE,GALLABRAT,DISTRIBUT
ION,DENY,EUCALYPTUS,WEDNESDAY,HAEMOGLOBIN,GLYCOGEN,ALKANE,BENZENE,CENTRIPETAL,OI
CYCLE,CHRISTMAS,CACTII,LLAMA,THIS FOOLED YOU,FLAW,QUEUE,OUAY,COMPLIMENTARY
2006 DATASUPPLEMENTARY,GYNAECOLOGIST,ESPY,ESTUARY,TEPID,CREEK,RIVER,CANYON,NITRO
GLYCERINE,TOLUENE
2007 DATACENTRIFUGAL,PROLONGATE,MENDACIOUS,AUDACIOUS,INTEGRATED,INTERNATIONAL,PL
ANETARY,VICARIOUS,MORON,TRANSITION,PERIODICALLY,APPLICATIONS,RELATIVITY,ULTRAVIO
LET,INFRARED,THEORETICAL,CATHODE,FOUNDATION,UNIVERSITY,EDUCATIONAL,REFERENCE,GNU
2008 DATAMICRO 80,STATIONARY,INERTIAL,ACCELERATION,END

```

# HOUSEHOLD ACCOUNTS PROGRAM FOR EXATRON STRINGY FLOPPY

## MODULE 1

```

5 POKE16634,74
10 CLS: CLEAR 7000:DEFDBLL:DEFSTRA:DEFINTI,J,X:DIMA(101):DIMLB(101):DIMLT(101)
15 POKE14396,23
20 F1$="DATE REF NO DETAILS TOTAL...A/C NO DEBIT...CREDIT.."
40 F3$="#####.##":F4$="£,###,###,##-"
45 @LOAD2

```

MODULE 2

```

50 PRINT"          * * * M E N U * * *"
60 PRINT"* 1 *  KEYBOARD INPUT","* 6 *  PRINT JOURNALS"
70 PRINT"* 2 *  CASSETTE INPUT","* 7 *  PRINT LEDGER BALANCES"
80 PRINT"* 3 *  READ MEMORY","* 8 *  LINEPRINTER UTILITY"
90 PRINT"* 4 *  EDIT MEMORY","* 9 *  LEDGER ACCOUNTS"
100 PRINT"* 5 *  SAVE ON CASSETTE"
150 X$="";X$=INKEY$:X=VAL(X$):IFX$=""THEN150
160 ON X GOTO 170,230,280,340,360,500,1010,1260,1300
170 @LOAD3
230 @LOAD4
280 @LOAD5
340 @LOAD6
360 @LOAD7
500 @LOAD8
1010 @LOAD9
1260 @LOAD10
1300 @LOAD11

```

MODULE 3

```

170 CLS:PRINT"  DATE  REF NO  DETAILS          TOTAL... A/C NO DEBIT...CREDIT."
171 PRINT"  !.....!.....!.....!.....!.....!.....!"
180 FORI=1TO100
190 IFA(I)=""THEN210
200 A="":INPUT A:IF LEN(A)>62THEN205
203 IFA="EXIT"THEN220
204 GOTO206
205 PRINT"ERROR...ENTRY EXCEEDS MAXIMUM LENGTH":GOTO200
206 A(I)=A
210 NEXT
215 I=I-4
220 CLS:PRINT"END OF BATCH...MEMORY CONTAINS ";I;" RECORDS":II=I
225 @LOAD2

```

MODULE 4

```

220 PRINT"REMOVE PROGRAM WAFER NOW"
225 GOSUB300:IFI=0THEN225ELSEPRINT"THANK YOU"
226 PRINT"INSERT DATA WAFER NOW"
227 GOSUB300:IFI=1THEN227ELSEPRINT"THANK YOU"
230 INPUT"HOW MANY RECORDS ARE THERE ON ESF WAFER":II
235 X$="":INPUT"PRESS (ENTER) WHEN READY TO LOAD DATA ";X$
236 @OPEN1
240 FOR I=1TOIISTEP4
250 @INPUTA(I),A(I+1),A(I+2),A(I+3)
255 PRINTA(I):PRINTA(I+1):PRINTA(I+2):PRINTA(I+3)
260 NEXT
265 @CLOSE
270 CLS:PRINT"INPUT FROM ESF WAFER COMPLETE"
275 PRINT"REMOVE DATA WAFER NOW"
276 GOSUB300:IFI=0THEN276ELSEPRINT"THANK YOU"
277 PRINT"INSERT PROGRAM WAFER NOW"
278 GOSUB300:IFI=1THEN278ELSEPRINT"THANK YOU"
280 @LOAD2
300 I=INP(240)AND1:RETURN

```

MODULE 5

```

280 CLS
290 PRINT"CONTENTS OF MEMORY"
300 FOR I=1TO100
310 PRINT "(";I;")":PRINT A(I):IFA(I)=""THEN330
320 IFI=80RI=160RI=240RI=320RI=400RI=480RI=560RI=640RI=720RI=800RI=880RI=96THENF
ORTX=1TO250:NEXTTX
325 NEXTI
330 PRINT"CONTENTS OF MEMORY READ":INPUT"PRESS (ENTER) TO CONTINUE ";X:CLS
340 @LOAD2

```

MODULE 6

```

340 CLS:PRINT"MEMORY EDIT":INPUT"WHICH LINE NO DO YOU REQUIRE":I
350 PRINTSTRING$(10,92);"  CURRENT LINE UNDER REVIEW  ";STRING$(10,92):PRINT"  "
;A(I):PRINT"IF AMENDMENT REQUIRED RE-TYPE LINE...OTHERWISE HIT (ENTER)"
HIT (ENTER) TO VIEW THE NEXT LINE IN THE BUFFER
TYPE & ENTER 'EXIT' TO RETURN TO MENU":INPUT A(I)
351 Z=0:INPUT Z$:Z=VAL(Z$)
352 IFZ$="EXIT"THENZ$="":GOTO500
353 IFZ=0THEN I=I+1:GOTO350
354 IFZ()0THEN I=Z:GOTO350
500 @LOAD2

```

MODULE 7

```

360 PRINT"REMOVE PROGRAM WAFER"
361 GOSUB450:IFI=0THEN361ELSEPRINT"THANK YOU"
362 PRINT"INSERT DATA WAFER"
363 GOSUB450:IFI=1THEN363ELSEPRINT"THANK YOU"
364 @OPEN1
370 FOR I=1TO100STEP4
380 @PRINTA(I):A(I+1),A(I+2),A(I+3)
385 PRINTA(I):PRINTA(I+1):PRINTA(I+2):PRINTA(I+3)
390 IF A(I)=""THEN:05
400 NEXT
405 I=I-4
406 @CLOSE
410 PRINT"RECORDING COMPLETE...THERE ARE ";I;" RECORDS ON WAFER":GOTO500
450 I=INP(240)AND1:RETURN
500 PRINT"REMOVE DATA WAFER"
505 GOSUB450:IFI=0THEN505ELSEPRINT"THANK YOU"
510 PRINT"INSERT PROGRAM WAFER"
515 GOSUB450:IFI=1THEN515ELSEPRINT"THANK YOU"
520 @LOAD2

```

MODULE 8

```

480 F1$="DATE REF NO DETAILS TOTAL...A/C NO DEBIT...CREDIT...":F3$="
@GCE,EE":F4$=
"E,EEE,EEE,EE-"
490 INPUT"(ENTER)":GO
500 PRINT:PRINT:PRINT" * * * JOURNALS AVAILABLE * * *"
510 PRINT"* 1 * PRINTOUT OF MEMORY"
520 PRINT"* 2 * CP CASH PAYMENTS JOURNAL"
530 PRINT"* 3 * CR CASH RECEIVED JOURNAL"
540 PRINT"* 4 * GJ GENERAL JOURNAL"
550 PRINT"* 5 * SJ SALES JOURNAL"
560 PRINT"* 6 * RETURN TO MAIN MENU"
570 X$="":X$=INKEY$:X=VAL(X$):IFX$=""THEN570
580 ON X GOTO 590,660,660,660,660,1500
590 CLS:PRINT"PRINTOUT OF MEMORY"
600 FOR I=1TO100
610 IFA(I)=""THEN640
620 PRINT(";I;"):PRINT A(I)
630 @PRINT(";I;"): " ":A(I)
640 NEXT
650 PRINT"PRINTOUT COMPLETE":LPRINT"PRINTOUT COMPLETE":GOTO500
660 INPUT"IS THE LINEPRINTER REQUIRED (Y/N)":P$
670 IFP$()="Y" AND P$()="N" THEN660
670 INPUT"WHICH DATE DO YOU REQUIRE "":E$
680 GOSUB900
690 FOR I=1TO100
700 IF A(I)=""THEN780
710 IF E$()LEFT$(A(I),6) AND KA$()MID$(A(I),40,2) THEN780
720 PRINTA(I):IFP$="N"THEN750
730 IFP$="Y"THEN LPRINT A(I)
750 DA$=MID$(A(I),31,8):DAE=VAL(DA$):OGE=OGE+DAE
760 DR$=MID$(A(I),47,8):DRE=VAL(DR$):OTE=DTE+DRE
770 CR$=MID$(A(I),55,8):CRE=VAL(CR$):CTE=CTE+CRE:BLE=BLE+DRE-CRE
780 NEXT
790 PRINT E$;" TOTALS":PRINTTAB(31)USINGF4$;OGE::PRINTTAB(47)USINGF3$;DTE;CTE
800 IFP$="Y"THEN LPRINT E$;" TOTALS":LPRINTTAB(31)USINGF4$;OGE::LPRINTTAB(47,
USINGF3$;DTE;CTE
810 OGE=0:OTE=0:CTE=0:GOTO490
820 INPUT"IS THE PRINTER REQUIRED (Y/N)":P$
830 IFP$()="Y" AND P$()="N" THEN820 ELSE RETURN
900 ON X-1 GOTO910,940,960,980
910 CLS:PRINT"CASH PAYMENTS JOURNAL FOR (DATE)":E$:PRINT F1$
920 IF P$="Y"THEN LPRINT"CASH PAYMENTS JOURNAL FOR (DATE)":E$:LPRINT F1$
925 KA$="CP":GOTO1000
940 CLS:PRINT"CASH RECEIVED JOURNAL FOR (DATE)":E$:PRINT F1$
950 IFP$="Y"THEN LPRINT"CASH RECEIVED JOURNAL FOR (DATE)":E$:LPRINT F1$
955 KA$="CR":GOTO1000
960 CLS:PRINT"GENERAL JOURNAL FOR (DATE)":E$:PRINTF1$
970 IFP$="Y"THEN LPRINT"GENERAL JOURNAL FOR (DATE)":E$:LPRINTF1$
975 KA$="GJ":GOTO1000
980 CLS:PRINT"SALES JOURNAL FOR (DATE)":E$:PRINTF1$
990 IFP$="Y"THEN LPRINT"SALES JOURNAL FOR (DATE)":E$:LPRINTF1$
995 KA$="SJ"
1000 RETURN
1500 @LOAD2

```

MODULE 9

```

800 GOTO1010
820 INPUT"IS THE PRINTER REQUIRED (Y/N) ";P$
830 IFP$()="Y" AND P$()="N" THEN820 ELSE RETURN
1010 CLS
1011 F4$="£,£££,£££.££-"
1020 PRINT"* * * LEDGER BALANCE OPTIONS * * *"
1030 PRINT"* 1 * DISSECTION SUMMARY THIS BATCH"
1040 PRINT"* 2 * RETURN TO MAIN MENU"
1080 X$="";X$=INKEY$:X=VAL(X$):IF X$=""THEN1080
1090 ON X GOTO 1100,1500
1095 COSUB820
1100 FOR J=1TO100:LB(J)=0:NEXTJ
1110 FOR I=1TO100
1120 J=VAL(MID$(A(I),42,2)):L1=VAL(MID$(A(I),47,8)):L2=VAL(MID$(A(I),55,8)):L=L1
-L2:L3(J)=LB(J)+L:LT(J)=LT(J)+LB(J):L=0:L1=0:L2=0
1140 NEXTI
1150 PRINT"DISSECTION TOTALS":IFP$="Y"THENLPRINT"DISSECTION TOTALS"
1160 FOR J=1TO100
1170 IF L3(J)=0THEN1200
1180 PRINTJ,:PRINTTAB(16)USINGF4$:LB(J):L=L+LB(J)
1190 IFP$="Y"THENLPRINTCHR$(15)J,:LPRINTTAB(48)USINGF4$:LB(J):LPRINTCHR$(14)
1200 NEXTJ
1210 PRINT"TOTAL":PRINTTAB(16)USINGF4$:L
1220 IFP$="Y"THENLPRINT"TOTAL":LPRINTTAB(48)USINGF4$:L
1230 L=0:PRINT"END OF RUN"
1240 IFP$="Y"THENLPRINT"END OF RUN"
1250 INPUT"PRESS ENTER TO CONTINUE ";X:GOTO1010
1500 @LOAD2

```

MODULE 10

```

1260 CLS:PRINT:PRINT"LINEPRINTER UTILITY
(TYPE HEADINGS OR NOTES AS REQUIRED - TYPE 'EXIT' TO RETURN TO
THE MAIN MENU)"
1270 M$="":INPUTM$:IFM$="EXIT"THEN1500
1280 PRINTM$:LPRINTM$
1290 GOTO1270
1500 @LOAD2

```

MODULE 11

```

1300 PRINT"          * * * LEDGER ACCOUNTS * * *"
1305 F1$="DATE  REF NO  DETAILS          TOTAL...A/C NO  DEBIT...CREDIT...":F4$="
£,£££,£££.££-"
1310 INPUT"IS THE LINEPRINTER REQUIRED (Y/N) ";P$
1320 IFP$()="Y"ANDP$()="N"THEN1310
1330 INPUT"WHICH ACCOUNT NO DO YOU REQUIRE ?
(ENTER '999' TO EXIT AND RETURN TO MENU) ";N
1350 IFN=1ANDN<=100GOTO1370
1360 IFN=999THEN1600ELSE1330
1370 PRINTF1$:IFP$="Y"THENLPRINTF1$
1380 FORI=1TO100:IFA(I)=""THEN1440
1390 IFN()VAL(MID$(A(I),42,2))THEN1440
1400 PRINTA(I):IFP$="Y"LPRTA(I)
1420 DR£=VAL(MID$(A(I),47,8)):CR£=VAL(MID$(A(I),55,8)):DT£=DT£+DR£:CT£=CT£+CR£:C
T£=CT£+DR£-CR£
1440 NEXT
1450 PRINT"TOTALS.....":PRINTTAB(38)USINGF4$:DT£:CT£
1460 IFP$="Y"LPRT"TOTALS.....":LPRINTTAB(38)USING F4$:DT£:CT£
1470 PRINT"BALANCE OF ACCOUNT.....":PRINTTAB(38)USINGF4$:CT£
1480 IFP$="Y"LPRT"BALANCE OF ACCOUNT.....":LPRINTTAB(38)USING F4$:CT£
1490 CT£=0:DT£=0:CT£=0:PRINT:PRINT:IFP$="Y"LPRT" ":LPRINT" "
1500 GOTO1330
1600 @LOAD2

```



\*\*\*\*\* NEXT MONTH'S ISSUE \*\*\*\*\*

Next month's issue will contain at least the following programs:

ESCAPEE (L1)

Interpret the computers clues accurately and you can find the escapee hidden behind one of the pillars.

CUP 80 (L1 or L2)

A simple simulation of a multi-horse race which demonstrates careful use of the '80 Random generator. L1 and L2 versions will be published.

LOTTO PREDICTOR (L2)

Combines the laws of statistical probability with the rules of chance and probability to predict next week's draw. You build up a data base, on disk or cassette, week by week and improve your chances.

THE WORLD (L1 or L2)

This program proves that you can get reasonable graphics on the '80. It draws a realistic representation of the world.

KB FIX

Especially with System 80 readers in mind, who may be experiencing difficulty in extracting copies of KB FIX from the opposition, here is a keyboard debounce routine, in BASIC which, on execution, loads a machine language routine into high memory which there permanently until switch -off. With to modify it to fit different memory size machines.

TRIANGLES

This program will be of interest to Architects, Engineers and Surveyors. Given three characteristics of ANY triangle (not just right angled triangles), it will calculate all the other characteristics!

Plus all the usual articles, letters, news etc..

Continued from P. 10 (Chess War)

I have not yet made mention of another chess cassette program available for micro-computers in Australia, Microchess. This is also a big seller and as it costs about the same as Sargon, many readers may be unsure of their choice. I have tested Microchess 1.5 and 2.0 and I must advise readers that Sargon is the market leader. I would rate Microchess 2.0 at about 1300 at the 3 minute level - slightly inferior to Voice Challenger. Anybody contemplating adding a Chess cassette program to their library would be well advised to purchase Sargon 2.5 or wait a little longer if they cannot obtain it at present.

tick where appropriate

To MICRO-80  
Please consider the enclosed program for...

( ) Publication in MICRO-80  
( ) Publication on disk or cassette only  
( ) Both

Name.....  
Address.....  
.....  
.....Post Code.....

===== CHECK LIST =====

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level 1, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it...etc

For system tapes, the start, end, and entry points, etc

The changes or improvements that you think may improve it.

Please package securely - padlocks are suggested - and enclose stamps or postage if you want your cassette or disk returned.



TO  
SUBSCRIBERS OF

# MICRO 30

**FOR THE FIRST TIME IN AUSTRALIA!  
GENUINE TANDY TRS-80 MICRO-COMPUTERS,  
HARDWARE AND SOFTWARE AT**

# GREAT DISCOUNT PRICES!!!

## HOW TO ORDER -

1. SELECT ITEMS FROM 1980 TANDY CATALOGUE
2. DEDUCT 10% FROM ADVERTISED PRICES
3. POST US YOUR ORDER STATING DESCRIPTION  
CAT. No AND A CHEQUE OR MONEY ORDER.

WE WILL —

1. ATTEND TO YOUR ORDER WITHIN 7 DAYS
2. SUPPLY GOODS SELECTED FREIGHT FREE !
3. SEND ADVERTISING REGULARLY TO KEEP YOU INFORMED OF CURRENT SPECIALS !

- *subject to availability*



# CONQUEST ELECTRONICS

PTY. LTD

**212 KATOOMBA ST KATOOMBA N.S.W. 2780 PHONE (047) 82 2491**

✂

**to CONQUEST ELECTRONICS Pty. Ltd.**  
**212 Katoomba St. KATOOMBA 2780**

**Please supply —**

QTY.	CAT NO	DESC.	AOV. PRICE
<b>SUB TOTAL</b>			
<b>LESS 10%</b>			
<b>FIND CHEQUE FOR TOTAL</b>			

**SEND FREIGHT FREE TO**

**NAME** .....

**ADDRESS** .....

**P/ Code** .....

# MICRO-80

## LEVEL II ROM REFERENCE MANUAL

by Edwin Paay

Published by MICRO-80 PRODUCTS

Written by Eddy Paay, the LEVEL II ROM REFERENCE MANUAL is the most complete explanation of the Level II BASIC interpreter ever published.

Part 1 lists all the useful and usable ROM routines, describes their functions explains how to use them in your own machine language programs and notes the effect of each on the various Z 80 registers.

Part 1 also details the contents of system RAM and shows you how to intercept BASIC routines as they pass through system RAM. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory—the only restriction is your own imagination!

Part 2 gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements, etc. It also describes the various formats used for BASIC, SYSTEM and EDITOR/ASSEMBLER tapes. Each section is illustrated by sample programs which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

The LEVEL II ROM REFERENCE MANUAL is intended to be used by machine language programmers. It assumes a basic understanding of the Z 80 instruction set and some experience of Assembly Language programming. But BASIC programmers too will benefit from reading it. They will gain a much better insight into the functioning of the interpreter which should help them to write faster, more concise BASIC programs.

# MICRO-80